CONSTRAINED AND UNCONSTRAINED LOCALIZATION FOR AUTOMATED INSPECTION OF MARINE PROPELLERS

by

RICHARD ALAN JINKERSON

B. S. Electrical Engineering, University of New Mexico
(1982)

Submitted to the Department of
OCEAN ENGINEERING
in Partial Fulfillment of the Requirements for the Degrees of

NAVAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1991

A. Douglas Carmichael
Chairman, Departmental Graduate Committee
Department of Ocean Engineering

CONSTRAINED AND UNCONSTRAINED LOCALIZATION FOR AUTOMATED INSPECTION OF MARINE PROPELLERS

by

Richard Alan Jinkerson

Submitted to the Department of Ocean Engineering on May 10, 1991
in partial fulfillment of the requirements for the degree of Naval Engineer
and Master of Science in Mechanical Engineering

Abstract

This work addresses the problem of optimal positioning of a set of measured points with respect to an ideal design surface. Localization refers to the process of determining the rigid body translations and rotations which must be performed on the set of points to move those points into closest correspondence with the design surface. In unconstrained localization all points have equal effect on the determination of the rigid body transformation, while constrained localization allows a subset of the points to have stronger influence on the transformation.

The measured points in the context of this work refer to physical points in space that are obtained by direct measurement of a manufactured marine propeller blade. The ideal design surface refers to a surface description of the propeller blade provided by the blade designer. Given that the measured blade is manufactured from the design surface description, it is the task of localization to determine an optimal positioning that will bring the measured points of the manufactured surface as close as possible to the design surface. If the manufactured blade is repositioned in space according to the prescription of the localization transformation, it will have the closest possible correspondence to the original design. Direct benefits to the manufacturer may result from less wasted material in initial castings and better ability to program postcasting work through optimal positioning of the workpiece.

The constrained and unconstrained localization method is developed from a theoretical basis. Applications of the localization method are investigated with examples of propeller designs and inspection data obtained from blades that were manufactured from those designs. Experimental results demonstrate the capabilities of the method and its applicability to automated inspection.

Thesis Supervisor:	Nicholas M. Patrikalakis
Title:	Associate Professor of Ocean Engineering
Thesis Reader:	David C. Gossard
Title:	Professor of Mechanical Engineering

Acknowledgements

This thesis is dedicated to my wife Karen. Without her support and encouragement this work would never have been completed.

I want to acknowledge the continued assistance provided to me by my thesis supervisor, Professor Nicholas Patrikalakis. He has been a wonderful mentor and friend, unsurpassed in the support of his students. I am very grateful for his expert guidance and patient assistance throughout this work.

I want to thank Dr. Franz-Erich Wolter and Dr. Nikiforos Papadakis, who were helpful to me in establishing some of the mathematical elements of the thesis and in developing a problem solution strategy. Both men are highly skilled in theoretical and applied mathematics; their expertise was valuable. I want to also thank Mr. Stephen Abrams, who is an exceptionally competent and diligent software engineer. He was able to transform my rudimentary computer programs into powerful and robust software of professional quality. Further, I wish to acknowledge Mr. Michael Drooker, Mr. Seamus Tuohy and Mr. Bradley Moran for their great assistance to me in the Ocean Engineering Design Laboratory at MIT.

I want to recognize Mr. Neal Holter of the Applied Research Laboratory at Pennsylvania State University, and Mr. Michael Koehler and Mr. Thomas York of Philadelphia Naval Shipyard. They provided the marine propeller designs and inspection data which were crucial to the experimental validation of the work presented in this thesis.

Finally I want to thank my thesis reader, Professor David Gossard. It was his enthusiastic personal style which first stimulated my interest in computer-aided design and computational geometry.

Funding for the Design Laboratory work in this area was obtained from the MIT Sea Grant College Program and the Naval Sea Systems Command of the U.S. Navy under contract numbers NA86AA-D-SG089 and NA16RG0093-01.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

A fundamental problem in manufacturing is the need to determine if a manufactured piece meets the requirements of the original design description from which it was made. The evaluation of positional tolerances to ensure that a manufactured item is an acceptable rendering of the original design is a basic element of manufacturing inspection.

In few areas of manufacturing inspection is the need for precise inspection and evaluation of tolerances more clearly demonstrated than in the area of marine propellers. The manufactured item is an exceedingly complex sculptured surface which must be produced with extremely high fidelity to the original design. Very strict positional tolerances must be achieved in a difficult manufacturing process to prevent severe compromise of the performance of the propeller.

The inspection of marine propellers has traditionally involved highly skilled technicians checking the surface of a manufactured propeller with numerous mechanical gages. The gages are cut to specified dimensions by the direction of a manufacturing engineer who interprets the specifications of the propeller designer. Although rigid guidelines are provided for placement of the gages on the blade, errors in measurement can result from decisions by the technician regarding "fit" or alignment of the gage on the manufactured blade. Moreover, the direct gage measurements only evaluate the blade at the local site where the measurement is made. A completely satisfactory method for evaluating global compliance with specified tolerances has not been available. Often expensive and

inefficient rework of propellers has been necessary because it was not possible to quickly and confidently ascertain whether the manufactured product would satisfy the requirements of the designer.

The recent development of automated methods of inspection using coordinate measuring machines (CMM) and laser-based measuring devices has made it possible to obtain voluminous quantities of highly accurate spatial measurements of manufactured propellers. These robotic devices have provided a reliable source of measurement data, but methods for best using that data are still being developed.

This thesis will address an aspect of the question of how to best utilize measured data from manufactured propellers for the automated inspection of those propellers. It will deal with the problem of optimal positioning of a set of measured points from a manufactured propeller blade relative to the design surface from which the blade was manufactured. The problem investigated in this thesis may be simply stated as follows:

> *Given a set of measured data points from a manufactured surface, determine the rigid body translations and rotations which must be applied to the set of measured data points to bring those points into closest correspondence with the design surface from which the measured surface was manufactured.*
>
> *If all measured points contribute equally to the determination of the set of rigid body transformations, then the problem is one of unconstrained localization.*
>
> *If some measured points have greater effect on the determination of the set of rigid body transformations than other points, then the problem is one of constrained localization.*

The thesis will develop the investigation of this problem using the following structure.

Chapter 2 will present a review of current literature relevant to the problem of localization. Particular emphasis will be given to literature which is directly related to the development of the localization algorithms presented in this thesis.

Chapter 3 will discuss the theoretical basis for the unconstrained localization algorithm. It will describe the optimization problem and the procedures that are followed to obtain a rigid body transformation matrix which is a solution to the unconstrained localization problem.

Chapter 4 will develop the theoretical basis for the constrained localization algorithm. It will contrast the differences between constrained and unconstrained localization and will describe the procedures involved in the solution of the constrained problem.

Chapter 5 will present experimental results which demonstrate the applicability to the problem of localizing manufactured marine propellers. One application involves the localization of a simple fan blade, while another application involves the localization of a complex marine propeller blade. In each case results are presented to demonstrate the usefulness of the constrained and unconstrained localization methods.

Chapter 6 summarizes the results of the investigation and presents ideas for additional work.

Appendix A will demonstrate the orthogonality of the rotational transformation matrix that is used in this investigation.

Appendix B will derive the Jacobian matrix for the objective function used in the unconstrained localization algorithm.

Appendix C will use orthogonality to develop a method for determining the parameters in a design surface of the projection of a point onto that surface to a very high degree of accuracy. The method finds direct application in improving the accuracy of the calculation of the minimum distance from a point to a parametric surface.

Chapter 2

LITERATURE REVIEW

There has been much interest in the problem of localization in recent years and the current literature reveals some aspects of methods that have been used in developing solutions to the problem. This chapter will review some of the literature that is relevant to the localization problem and the work of this investigation. The intent is to provide some pertinent background information which will give the reader a broader perspective concerning this particular work.

Localization of surfaces was accomplished by Thorne using a least squares matching of associated boundary edges [Thorne 85]. Gunnarsson developed a localization method between a set of points and parametric surfaces by dynamically faceting the design surface and finding a rigid body transformation matrix which minimizes the sum of squared distances from the data points to associated planar faces [Gunnarsson 87a], [Gunnarsson 87b]. This formulation required the solution of a constrained minimization problem with 12 unknowns, which are the elements of the rotation matrix with three translations, and 6 constraints which are the necessary relations between elements of the Euclidean rotation matrix.

One question that is intrinsic to the formulation of the localization problem is that of selection of an appropriate norm to use in distance minimization. Bourdet and Clement present an analysis of this problem in [Bourdet 88]. They show, through the use of a small-displacement screw linearization model, that the L_∞ (or minimax) norm reduces form error when compared to the L_2 (or least-squares) norm by approximately 15 per cent for a

small number of points. The improvement however disappears when the number of points is greater than twenty. In contrast, the L_2 norm is superior to the L_∞ norm in its usefulness for statistically detecting aberrant points in a particular data set. Moreover, the L_∞ norm was shown to require much greater computational time than the L_2 norm. The general guidelines of the paper can be summarized by saying that for point sets of fewer than 12 points the L_∞ norm should be used, while the L_2 norm should be used for point sets of greater than 12 points. An interesting algorithm which allows for computation of both the L_2 and the L_∞ norms at the same time is presented in [Goch 90].

Some of the theoretical framework upon which this work is based is presented by Alt, Mehlhorn, Wagener and Welzl in [Alt 88]. The writers of the paper demonstrate some algorithms which pertain to the problem of mapping congruent objects A, B in $\Re^n$ from one to the other using geometric transformations involving rotations, translation, reflections, and stretching. Although the algorithms developed in this thesis do not directly address reflection and stretching, the mapping problem is essentially the same one. A significant finding in [Alt 88] is that exact congruencies are not possible using measured data. Even small perturbations in the measurements will destroy conguencies between the two geometries. They therefore establish the *approximate congruence problem with tolerance* ε, where the maximum distance between two corresponding sets of points is less than or equal to ε. This problem is shown to have a computational upper bound of $O(n^3 \log n)$ for either L_2 or L_∞ norms, where n is the number of points, when the sets of points are assumed to be known and a congruency relationship exists between the two point sets. The writers of the paper deal only with point sets; they do not discuss the particular problem of mapping approximately congruent point sets to a *surface*.

Imai, Sumino and Imai continue work on the problem of mapping point sets in [Imai 88]. They develop a minimization algorithm which greatly improves the theoretical efficiency of the methods of Alt, et al. They show that, by using their algorithm, the L_2 norm problem has a computational upper bound of $O(n^3)$ and the L_∞ norm problem has a computational upper bound of $O(n^2 + n \log n)$ in the worst case, where n is again the number of points.

An essential element of the localization algorithms presented in this thesis is the use of orthogonal projection methods of curves onto surfaces to locate nearest points in the parametric space of a design surface. These methods were developed originally by Pegna and Wolter in [Pegna 90]. In this paper a space curve is mapped onto a surface by tracing a surface curve whose points are connected to the space curve by surface normals. The tracing is achieved by solving a tensorial differential equation in the parametric space of the surface.

Another concept that is important in this work was developed by Kriezis, Patrikalakis and Wolter in [Kriezis 90], [Kriezis 91] for use in the solution of surface intersection problems. It is the oriented distance function which is defined as the inner product between a vector from a given point to its nearest point on a surface, and the unit normal vector of the surface at that point. The method is used in formulating the constrained localization problem of this investigation under the assumption that the two vectors are collinear.

The fundamental basis for the work of this thesis applied to the unconstrained localization problem derives from earlier work presented by Bardis and Patrikalakis in [Patrikalakis 90]. In this work positional tolerances were represented in terms of a ball offset tolerance region around an ideal rational spline surface (design surface). The tolerance region bounding surfaces were approximated by rational B-splines. The manufactured surface (target surface), known either in terms of a lattice of measured points or as a

simulated surface from numerical or analytical predictions, was then optimally positioned with respect to the ideal design surface by minimizing the L_2 distance norm. After localization, the target surface could be tested for intersection with the boundaries of the tolerance region. If the target surface is found to be entirely within this tolerance region, then the agreement would be considered satisfactory. These unconstrained localization techniques were further developed to improve their computational efficiency, and they are presented in a paper coauthored by this writer [Bardis 91].

Chapter 3

UNCONSTRAINED LOCALIZATION

3.1 Introduction

Localization can be defined as *the problem of determining the optimal positioning of a set of measured points relative to a design surface.* If all measured data points have equal effect on the determination of this optimal positioning, then the localization problem can be defined as a problem of **unconstrained** localization.

The localization problem can be formulated as an optimal parameter estimation problem involving six parameters. Those six parameters are the three translations and three Euler angles which correspond to a general three-dimensional translation and rotation of a rigid body in space. The problem can be formulated as an unconstrained minimization, where the objective function of the minimization is the sum of squared minimum distances of a set of measured points from a design surface. In this context the measured points represent physical points in space that are determined by direct measurement of a manufactured surface. The design surface refers to the underlying design description that is used to produce the manufactured surface. The unconstrained localization problem is then the problem of estimating the six parameters of a rigid body transformation which will bring the set of equally weighted measured points into closest correspondence with the design surface.

3.2 Problem Formulation

Consider a parametric surface $P(u,v)$, which will be called the **design surface**, representing the desired design geometry. Consider also a set of m points $\mathbf{R}_i$, $1 \le i \le m$, $\mathbf{R}_i \in \Re^3$, which will be called **measured points**. Finally, consider another set of m points $\mathbf{Q}_i$, $1 \le i \le m$, in the design surface $P(u,v)$ which are the nearest points to each measured point $\mathbf{R}_i$. It is assumed that the nearest points $\mathbf{Q}_i$ are unique and are the orthogonal projections of the points $\mathbf{R}_i$ onto $P(u,v)$. The points $\mathbf{Q}_i$ will be subsequently called **projections.**

The assumption that the nearest points $\mathbf{Q}_i$ are unique and also orthogonal projections of the points $\mathbf{R}_i$ onto $P(u,v)$ is reasonable for the inspection problem that is being investigated. The measured points of a manufactured propeller blade can be expected to be very close to the surface of the propeller and relatively far from the medial axis of the blade. It can be shown that under some reasonable conditions, the nearest point will be unique [Kriezis 91]. Furthermore, the only places on the blade where the orthogonal projection condition could be expected to not be valid would be at the trailing edge, root edge or tip of the blade, and these locations are ones where measurements cannot be readily made anyway because of the edge discontinuity.[1]

The objective function for minimization is the sum of squared minimum distances of each point from the design surface. Each measured point $\mathbf{R}_i$ has a nearest point $\mathbf{Q}_i$ on the design surface. The minimum distance from each measured point to the design

[1] Even though the assumption of the existence of a unique nearest point which is also an orthogonal projection of a measured point to the design surface is not unrealistic for this problem, its importance cannot be overemphasized. It forms a foundation for much of the theoretical development of this thesis.

surface can be simply defined as the Euclidean distance between the points $\mathbf{R}_i$ and $\mathbf{Q}_i$. If the distance between two points is denoted by $d(\mathbf{P}_1, \mathbf{P}_2)$ then the minimum distance from a measured point $\mathbf{R}_i$ to the design surface $\mathbf{P}(u, v)$ is defined as

$$d(\mathbf{R}_i, \mathbf{Q}_i) = | \mathbf{R}_i - \mathbf{Q}_i | = \min_{u,v} d\ [\mathbf{R}_i, \mathbf{P}(u, v)] \tag{3-1}$$

The squared minimum distance is

$$(\mathbf{R}_i - \mathbf{Q}_i) \cdot (\mathbf{R}_i - \mathbf{Q}_i) = [d(\mathbf{R}_i, \mathbf{Q}_i)]^2 \tag{3-2}$$

and the objective function becomes

$$OF = \sum_{i=1}^{m} [d(\mathbf{R}_i, \mathbf{Q}_i)]^2 \tag{3-3}$$

The parametric surface is assumed to be an untrimmed rational B-spline (NURBS) patch of orders M in u, and N in v, $(0 \le u, v \le 1)$. The surface is further assumed to have continuous first order partial derivatives (C^1 continuity). The design surface patch is then of the form

$$\mathbf{P}_{M,N}(u, v) = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} h_{ij} \mathbf{P}_{i,j} B_{i,M}(u) B_{j,N}(v)}{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} h_{ij} B_{i,M}(u) B_{j,N}(v)} \tag{3-4}$$

$\mathbf{P}_{i,j}$, $(0 \le i \le m-1,\ 0 \le j \le n-1)$ are the vertices of the associated control polyhedron and h_{ij}, $(0 \le i \le m-1,\ 0 \le j \le n-1)$ are positive weights. $B_{i,M}(u)$ and $B_{j,N}(v)$ are the B-spline basis functions over open knot vectors [Gordon 74] with variable knot spacing of

$$U = (u_0, u_1, \ldots, u_{m+M-1}) \tag{3-5}$$

$$V = (v_0, v_1, \ldots, v_{n+N-1}) \tag{3-6}$$

The goal of the localization problem is to obtain an optimal rigid body transformation for operation on the set of measured points, $\mathbf{R}_i$, so that those points will correspond as closely as possible to the design surface, $\mathbf{P}(u,v)$.

It is possible to generalize the objective function to accomodate multiple untrimmed patches using analogs of the equations for a single patch. Consider n patches such that $\mathbf{P}_j(u,v)$, $1 \le j \le n$, represents the jth patch, and $\mathbf{R}_{ij}$, $1 \le i \le m_j$, $1 \le j \le n$, $\mathbf{R}_{ij} \in \Re^3$, represents the ith measured point associated with the jth patch. If $\mathbf{Q}_{ij}$ now represents the minimum distance orthogonal projection of $\mathbf{R}_{ij}$ onto the design surface $\mathbf{P}_j(u,v)$, then the squared minimum distance for multiple patches becomes

$$(\mathbf{R}_{ij} - \mathbf{Q}_{ij}) \cdot (\mathbf{R}_{ij} - \mathbf{Q}_{ij}) = [d(\mathbf{R}_{ij}, \mathbf{Q}_{ij})]^2 \tag{3-7}$$

and the objective function for multiple patches becomes

$$OF = \sum_{j=1}^{n} \sum_{i=1}^{m_j} [d(\mathbf{R}_{ij}, \mathbf{Q}_{ij})]^2 \tag{3-8}$$

3.3 Localization Transformation

The localization procedure that is used to determine the optimum rigid body transformation for the measured points consists of minimizing the objective function of (3-3) by calculating values for the *three rotations* and *three translations* which are the six parameters of a rigid body transformation.

The three rotational parameters of the rigid body transformation are given by the angles ψ, θ and φ (Euler angles) which represent rotations about the x, y and z axes

respectively. The three translational parameters are represented as elements of a translation vector **t**, having components t_x, t_y and t_z for translations along each respective axis.

If the set of measured points of the localization problem is operated upon by successive rotations followed by successive translations, then a new set of points, $\mathbf{r}_i$, can be obtained from the original measured points by the operation

$$\mathbf{r}_i = [C]\mathbf{R}_i + \mathbf{t} \tag{3-9}$$

In equation (3-6) the matrix $[C]$ is defined as the **rotational transformation matrix**, which is obtained by multiplying the matrices associated with rotations about each coordinate axis. If rotations are ordered in the sequence rotation by angle φ about the z-axis, followed by rotation by angle θ about the y-axis, followed finally by rotation by angle ψ about the x-axis, then the matrix $[C]$ is given by

$$[C] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3-10}$$

or

$$[C] = \begin{pmatrix} \cos\theta\cos\phi & -\cos\theta\sin\phi & \sin\theta \\ \cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi & \cos\psi\cos\phi - \sin\theta\sin\phi\sin\psi & -\sin\psi\cos\theta \\ \sin\psi\sin\phi - \sin\theta\cos\phi\cos\psi & \sin\psi\cos\phi + \sin\theta\sin\phi\cos\psi & \cos\theta\cos\psi \end{pmatrix} \tag{3-11}$$

The matrix $[C]$ is an orthogonal matrix, having the property $[C][C]^T = [C]^T[C] = I_3$, where I_3 is the identity matrix of dimension 3.[2]

[2] A demonstration showing that $[C]$ is orthogonal can be found in Appendix A.

3.4 Localization Algorithm

The localization algorithm is the unconstrained minimization of the objective function given as equation (3-3). The process involves iterative operations on the set of measured points $\mathbf{R}_i$, producing new sets of transformed points according to equation (3-9). If $\mathbf{r}_i$ is a transformed point given by equation (3-9), and $\mathbf{q}_i$ is the minimum distance orthogonal projection of the transformed point onto the design surface, then the minimum distance squared from $\mathbf{r}_i$ to $\mathbf{P}(u,v)$, $[d(\mathbf{r}_i, \mathbf{q}_i)]^2$, is defined by the following equation which follows directly from equation (3-2):

$$[d(\mathbf{r}_i, \mathbf{q}_i)]^2 = (\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i) \tag{3-12}$$

Using the transformed points $\mathbf{r}_i$ and the corresponding projections $\mathbf{q}_i$ to calculate the squared minimum distance at each iteration step, the iterative process continues until a minimum is reached. This minimization can be simply stated as the following:

Determine $\phi, \theta, \psi, t_x, t_y, t_z$, such that

$$OF(\phi, \theta, \psi, t_x, t_y, t_z) = \sum_{i=1}^{m} [d(\mathbf{r}_i, \mathbf{q}_i)]^2 \tag{3-13}$$

is minimized.

An alternative objective function would be the minimax (or Tschebyscheff norm), L_∞ , where

$$OF(\phi, \theta, \psi, t_x, t_y, t_z) = max_i \, \| \text{Sup } [d(\mathbf{r}_i, \mathbf{q}_i)] - \text{Inf } [d(\mathbf{r}_i, \mathbf{q}_i)] \| \tag{3-14}$$

is minimized instead of equation (3-13). Such a norm allows calculation of the parameters minimizing the maximum of the minimum distance of all measured points from the design surface. However an objective function of this type is much harder to

implement for a large number of points, and [Bourdet 88] has shown that no significant improvement over the L_2 norm results if the number of measured points is greater than about 20.

Determination of the minimum distance from a point to a parametric surface using (3-1) involves the calculation of a minimum with respect to the design surface parameters u and v. A modified Newton algorithm implemented in the Numerical Algorithm Group routine E04KCF, [Gill 76], [NAG 89], is used with an initial guess of the minimum distance position provided by using the u, v parameters of an orthogonal projection onto $\mathbf{P}_{M,N}(u,v)$ of B-spline curves that are fitted through selected sets of data points. This method uses the orthogonal projection techniques of curves on surfaces developed in [Pegna 90]. It has resulted in very good computational efficiency for two reasons:

1. An exhaustive search of $\mathbf{P}_{M,N}(u,v)$ to compute minimum distance needs to be performed only once for the end point of each B-spline curve to find its minimum distance orthogonal projection.
2. The B-spline approximation for the measured data can be very rough.

The modified Newton algorithm used for calculation of the distance of each point to the design surface behaves well if $\mathbf{r}_i$ is close to $\mathbf{P}_{M,N}(u,v)$. For the small rotations and translations that are performed on each point using (3-9) it is unlikely that several local minima of $d(\mathbf{r}_i,\ \mathbf{P}_{M,N}(u,v))$ will interfere with the process.

The unknown quantities $\phi, \theta, \psi, t_x, t_y, t_z$ which render the objective function (3-2) minimum are computed by applying a quasi-Newton algorithm implemented in Numerical Algorithms Group routine E04JAF. Estimates of the Jacobian and Hessian of

$OF(\phi, \theta, \psi, t_x, t_y, t_z)$ are used to generate a series of feasible sixtuples converging to a minimum, [Gill 74], [NAG 89].[3] The initial guess of the unknown quantities is chosen as the zero vector.

After one calculation of rotations and translations, new points $\mathbf{r}_i$ are created using equation (3-9). The process continues until a minimum for the objective function is achieved within a tolerance level specified by the user. At the conclusion of the process the user will be provided with an optimal prescription for the three rotations and three translations which should be applied to the measured points to bring them into closest correspondence with the design surface.

[3] The Jacobian of the objective function is estimated in the NAG routine E04JAF by difference quotients. Direct calculation of the Jacobian might however improve the performance and accuracy of the quasi-Newton algorithm if implemented in another routine. The derivation of the Jacobian for the objective function is therefore provided in Appendix B.

Chapter 4

CONSTRAINED LOCALIZATION

4.1 Introduction

Unconstrained localization addresses the problem of determining the optimal positioning of a set of measured points relative to a design surface when each measured point has an equal effect on the determination of positioning. **Constrained localization,** on the other hand, involves *the problem of determining a feasible, but not necessarily optimal, positioning of a set of measured points relative to a design surface when subsets of the measured points can have unequal effects on the determination of positioning.*

The unconstrained localization problem seeks to minimize one global objective function so that measured points are all collectively brought as close to the design surface as possible. In this problem each point contributes with the same weight to the minimization of the objective function. In contrast, the constrained localization problem starts with the rotation and translation produced by the minimized objective function of the unconstrained localization problem and determines a rigid body transformation which will allow the measured points to satisfy a set of nonlinear constraints. The constrained localization problem does not minimize an objective function, but rather uses minimization techniques to find a feasible transformation that will satisfy the constraints imposed by a set of constraint functions. Satisfying the constraint functions has the effect of changing the importance of each measured point.

The need for constrained localization of marine propellers is evident because of the tighter positional tolerances that are required near the leading edge of a blade. A method is required which will provide for greater influence on the localization by measured points that are close to the leading edge.

The typical design of a propeller blade provides a convenient way to construct the constrained localization problem. Normally the blade is described by the designer in terms of a NURBS surface with two sets of isoparameter lines running approximately parallel and perpendicular to the leading edge of the blade. In this manner the isoparameter lines form a grid in the spanwise and chordwise directions of the blade. Two spanwise isoparameter lines can be assigned to bound the leading edge region of the blade. Constraints on the localization can then be imposed based upon whether or not parametric values of the minimum distance orthogonal projections of measured points on the design surface lie inside or outside of the boundaries defined by the two isoparameter lines. If the points within the two isoparameter lines in the parametric space of the surface are nearest points to the *measured* points, then measured points near the leading edge of the blade can have greater effect upon the localized positioning as required. This is accomplished by associating each measured point with its respective projection, and using the position of the projection in the parametric space of the design surface to determine the *constrained* function and consequent effect of the measured point upon the localization.

The constrained localization method assumes that the region of the design surface that is associated with each measured point by its projection on the design surface does not change during the localization. This assumption may not always be valid if the projections of measured points are very close to the isoparameter lines which bound the

leading edge region of the blade. For the small rotations and translations that are anticipated for this method, it is not expected that the projections of many measured points will change regions during the localization. Furthermore, even if such a phenomenon occurs the overall effects on the inspection problem are expected to be relatively small.

4.2 Problem Formulation

The constrained localization problem can be formulated as a nonlinear constrained optimization problem. The constraints are limits on distances of measured points from the design surface. They are determined by positions in the parametric space of the design surface of points which are nearest to measured points. In the formulation of this problem the constraints are inequality bounds on the constraint function which are allowed to have only two possible values.[4] If the points in the design surface lie within the isoparametric boundaries which define the leading edge of the blade, then the associated constraint on the optimization will have one value; if the points lie outside of this boundary, then the associated constraint will have another (larger) value. The problem is inhererently nonlinear because the rotation transformations which operate on the measured points to change their distances from the design surface are formed from combinations of transcendental functions in three independent variables.

[4] Permitting only two values for the constraints reduces the complexity of the problem, but may make it more difficult to solve because of the forced discontinuity at the boundary. An alternative formulation would provide a transitional *region* for the constraint function to mitigate the transition boundary problem.

4.2.1 Global Objective Function

It is assumed in the formulation of the constrained localization problem that the unconstrained localization problem has been previously solved using the techniques of Chapter 3. In this case the objective function given by

$$OF = \sum_{i=1}^{m} [d_p R_i]^2 \tag{4-1}$$

has already been minimized. This means that at the beginning of the constrained localization an optimal *global* positioning to place the set of measured points at the minimum distance from the design surface has already been determined. Given this initial condition there is no need to further minimize a global objective function. On the other hand, constraint functions for each measured point must be determined, and a solution procedure must be performed to find an appropriate positioning which will satisfy the constraints. Since it is assumed that the constrained localization starts from a position of global optimization, it is expected that the constrained localization procedure will not produce an improvement in the global result obtained from the unconstrained localization. *A satisfactory result will be to find a positioning of the measured points which satisfies the constraints.* This result will not be necessarily unique.

The assumptions associated with the global objective function for the constrained localization problem can be summarized in the following statements:

(1) The constrained localization problem starts from a position of global optimization which is the solution to the unconstrained localization.

(2) The constrained localization cannot improve the global result of the

unconstrained localization and therefore does not attempt a global optimization.

(3) A satisfactory result is a solution, not necessarily unique, which satisfies the constraints.

4.2.2 Constraint Function Definition

The selection of an appropriate constraint function is fundamental to the formulation of the constrained localization problem. The constraint function must certainly be a distance measure, but careful definition of this measure may facilitate the solution of the problem.

4.2.2.1 Squared Distance Function

Consider again the parametric design surface, $\mathbf{P}(u,v)$, the set of m measured points, $\mathbf{R}_i$, and the set of m unique nearest points $\mathbf{Q}_i$ on $\mathbf{P}(u,v)$, which are also assumed to be orthogonal projections. The minimum distance from a measured point $\mathbf{R}_i$ to the design surface $\mathbf{P}(u,v)$ was defined as

$$d(\mathbf{R}_i, \mathbf{Q}_i) = |\mathbf{R}_i - \mathbf{Q}_i| = \min_{u,v} d\,[\mathbf{R}_i, \mathbf{P}(u,v)] \tag{4-2}$$

and the minimum distance squared was given as

$$(\mathbf{R}_i - \mathbf{Q}_i) \cdot (\mathbf{R}_i - \mathbf{Q}_i) = [d(\mathbf{R}_i, \mathbf{Q}_i)]^2 \tag{4-3}$$

This function provides a measure of the proximity of a point to a surface and was chosen as the objective function for the unconstrained localization. However the squared distance function is not the best one to use for every type of problem. For minimization problems involving very small changes in magnitude, the squared function introduces inaccuracy because all changes are squared. This

effect is manifested by reducing the number of decimal places that can be calculated with confidence by half. (From another point of view, double precision calculations effectively become single precision calculations.) Moreover the squared distance function as an *unsigned* function causes all sense of position of a measured point relative to the design surface normal vector orientation to be lost. The loss of positional sense of a measured point relative to the design surface is particularly undesirable in the context of using the process as an inspection tool. A manufacturing engineer evaluating the results of a localization operation would want to know whether he should cut or weld a manufactured blade. Therefore positional sense is essential.

Good results were obtained for the unconstrained problem using the squared distance objective function, notwithstanding its previously mentioned shortcomings.[5] However it was judged at the beginning of the investigation of the constrained problem that the squared distance function might not provide sufficient numerical accuracy for solution of this more difficult nonlinear problem. An alternative distance function was desired specifically to enhance the numerical accuracy that could be expected from calculations.

4.2.2.2 Oriented Distance Function

A different method for determining the distance from a point to a surface using an **oriented distance** function was introduced in [Kriezis 90], [Kriezis 91].

[5] An expanded discussion of the results of the unconstrained localization is provided in Chapter 5.

This function does not use a squared quantity, and it retains the sense of relative position between a given point and a surface by using the normal of the nearest point in the surface to the given point.

If $\mathbf{n}_i$ is the unit normal vector of the design surface at the nearest point $\mathbf{Q}_i$ on $\mathbf{P}(u,v)$, which is also an orthogonal projection of the measured point $\mathbf{R}_i$ on $\mathbf{P}(u,v)$, then the **oriented distance**, $\hat{d}(\mathbf{R}, \mathbf{Q})$, from $\mathbf{R}_i$ to the design surface $\mathbf{P}(u,v)$ can be defined as

$$\hat{d}(\mathbf{R}_i, \mathbf{Q}_i) = \mathbf{n}_i \cdot (\mathbf{R}_i - \mathbf{Q}_i) \tag{4-4}$$

This function will produce a very accurate measure of the distance from a point to a surface if the difference vector $\mathbf{R}_i - \mathbf{Q}_i$ can be calculated with high accuracy. A method for improving the accuracy of this calculation by exploiting the orthogonality of the difference vector to the design surface tangent plane is developed in Appendix C.

The oriented distance function has a form similar to (4-2), but retains the positional sense of the unit normal vector $\mathbf{n}_i$ at each projection on the design surface.

4.2.2.3 Constraint Assignment

Using the oriented distance as the constraint function, the localization will have m constraints c_i, one for each measured point $\mathbf{R}_i$, defined as

$$c_i = \hat{d}(\mathbf{R}_i, \mathbf{Q}_i) \tag{4-5}$$

The constraints are assigned based upon the position of projections relative to the isoparameter lines which define the leading edge region of the blade using the following scheme. Assume the chordwise parametrization of the design surface in the parameter u with u_1, u_2 the isoparameter lines at the leading edge boundaries. Let ε be the value of the constraint for projections in the leading edge region and δ, the value of the constraint for other points. Typically, $\varepsilon \ll \delta$. If u_i is the u parameter in the design surface of the projection $\mathbf{Q}_i$, then constraints c_i, can be assigned to each measured point according to

$$-\varepsilon \le c_i \le +\varepsilon \tag{4-6}$$

if $u_1 \le u_i \le u_2$,

or

$$-\delta \le c_i \le +\delta \tag{4-7}$$

if $u_i < u_1$ or $u_i > u_2$

4.3 Constrained Localization Algorithm

The constrained localization is the problem of determing the rotations and translations which must be performed on the set of measured points so that they will satisfy the required localization constraints. The problem can be summarized in the following problem statement.

For a set of m measured points $\mathbf{R}_i$, having nearest point orthogonal projections $\mathbf{Q}_i$ on the parametric design surface $\mathbf{P}(u, v)$, where $1 \le i \le m$, determine the set of rigid body rotations and translations φ, θ, ψ, t_x, t_y, t_z such that the oriented distances from $\mathbf{R}_i$ to $\mathbf{Q}_i$ satisfy the constraints c_i as they are defined in (4-6) and (4-7)

As in the case of the unconstrained localization problem, the measured points are operated upon by successive rotations followed by successive translations to produce a new set of points $\mathbf{r}_i$, defined by

$$\mathbf{r}_i = [C]\mathbf{R}_i + \mathbf{t} \tag{4-8}$$

where $\mathbf{t}$ is again the translation vector with components t_x, t_y, t_z and $[C]$ is the rotational transformation matrix given by

$$[C] = \begin{pmatrix} \cos\theta\cos\phi & -\cos\theta\sin\phi & \sin\theta \\ \cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi & \cos\psi\cos\phi - \sin\theta\sin\phi\sin\psi & -\sin\psi\cos\theta \\ \sin\psi\sin\phi - \sin\theta\cos\phi\cos\psi & \sin\psi\cos\phi + \sin\theta\sin\phi\cos\psi & \cos\theta\cos\psi \end{pmatrix} \tag{4-9}$$

The problem of determining the set of six parameters which will allow the set of measured points to satisfy the localization constraints is solved using the routine E04UCF for nonlinear constrained optimization problems provided by the Numerical Algorithms Group (NAG). The routine uses an iterative sequential quadratic programming (SQP) algorithm in which the search direction is the solution of a quadratic programming (QP) problem, [Gill 86], [NAG 89].

The nonlinear constrained optimization routine estimates gradients of user-supplied functions with difference quotients unless the user can also supply those gradients. The latter situation produces a great improvement in computational accuracy and efficiency.

For this reason, part of the implementation of this algorithm involves providing symbolic gradients (or equivalently Jacobians), for each of the functions that are supplied to the NAG routine.

4.3.1 Global Objective Function

Although an objective function for the constrained localization is not minimized as previously explained, the structure of the nonlinear constrained optimization routine requires an objective function to be supplied. It is sufficient in this case to let the objective function be defined simply as a **constant**.[6] This is the simplest possible definition for an objective function and it allows the optimization routine to work accurately and rapidly. The objective function may be defined then as the following:

Determine $\phi, \theta, \psi, t_x, t_y, t_z$, such that

$$OF(\phi, \theta, \psi, t_x, t_y, t_z) = \text{Constant} \tag{4-10}$$

4.3.2 Objective Function Jacobian

The NAG routine E04UCF requires the Jacobian of the objective function to be supplied for most efficient operation. Since the objective function has been defined as a *constant*, the Jacobian, which is the first partial derivative of the objective function in each independent variable is identically equal to zero in all six variables.

---

[6] This simple but profound idea was first suggested by Dr. Nikiforos Papadakis of the MIT Ocean Engineering Design Laboratory. He has conducted extensive research in optimization methods.

So the Jacobian of the objective function, J_O, may be defined as

$$J_O = \begin{pmatrix} \frac{\partial OF}{\partial \phi} \\ \frac{\partial OF}{\partial \theta} \\ \frac{\partial OF}{\partial \psi} \\ \frac{\partial OF}{\partial t_x} \\ \frac{\partial OF}{\partial t_y} \\ \frac{\partial OF}{\partial t_z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{4-11}$$

Therefore the Jacobian for the constant objective function is the zero vector.

4.3.3 Constraint Function

The constraint function consists of inequality constraints on the oriented distance function given as

$$\hat{d}(\mathbf{R}_i, \mathbf{Q}_i) = \mathbf{n}_i \cdot (\mathbf{R}_i - \mathbf{Q}_i) \tag{4-12}$$

If $\mathbf{q}_i$ is the projection of a transformed measured point $\mathbf{r}_i$ defined by (4-8), then a new oriented distance function after a transformation operation will be given by

$$\hat{d}(\mathbf{r}_i, \mathbf{q}_i) = \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \tag{4-13}$$

At each iteration a new transformed measured point $\mathbf{r}_i$, and a new corresponding minimum distance projection $\mathbf{q}_i$, are determined for each measured point $\mathbf{R}_i$ existing before the transformation. This procedure produces a new oriented distance $\hat{d}(\mathbf{r}_i, \mathbf{q}_i)$ for each measured point at each iteration, using the most recent transformed point as a starting point for the computation of $\mathbf{q}_i$.

4.3.4 Constraint Function Jacobian

The Jacobian for the constraint functions are the values of the first partial derivatives of the functions in each of the independent variables of the problem. The definition of the oriented distance function as the constraint function for this problem allows these derivatives to be calculated in a straightforward manner.

Since the transformed oriented distance function is the constraint function for the localization problem during any particular iteration step, the determination of the Jacobian for this function consists of calculating a set of m first partial derivatives in the six variables φ, θ, ψ, t_x, t_y, and t_z. The Jacobian $\boldsymbol{J}_i$, $1 \le i \le m$, for this problem may thus be defined as

$$\boldsymbol{J}_i^T = \begin{pmatrix} \frac{\partial \hat{d}(\mathbf{r}_i, \mathbf{q}_i)}{\partial \phi} \\ \frac{\partial \hat{d}(\mathbf{r}_i, \mathbf{q}_i)}{\partial \theta} \\ \frac{\partial \hat{d}(\mathbf{r}_i, \mathbf{q}_i)}{\partial \psi} \\ \frac{\partial \hat{d}(\mathbf{r}_i, \mathbf{q}_i)}{\partial t_x} \\ \frac{\partial \hat{d}(\mathbf{r}_i, \mathbf{q}_i)}{\partial t_y} \\ \frac{\partial \hat{d}(\mathbf{r}_i, \mathbf{q}_i)}{\partial t_z} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial \phi} \\ \frac{\partial \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial \theta} \\ \frac{\partial \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial \psi} \\ \frac{\partial \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial t_x} \\ \frac{\partial \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial t_y} \\ \frac{\partial \mathbf{n}_i \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial t_z} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{n}_i \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)}{\partial \phi} \\ \frac{\partial \mathbf{n}_i \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)}{\partial \theta} \\ \frac{\partial \mathbf{n}_i \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)}{\partial \psi} \\ \frac{\partial \mathbf{n}_i \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)}{\partial t_x} \\ \frac{\partial \mathbf{n}_i \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)}{\partial t_y} \\ \frac{\partial \mathbf{n}_i \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)}{\partial t_z} \end{pmatrix} \tag{4-14}$$

which when the partial derivatives are expanded is equivalent to

$$J_i^T = \begin{pmatrix} \frac{\partial \mathbf{n}_i}{\partial \phi} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) + \mathbf{n}_i \cdot \left(\frac{\partial [C]}{\partial \phi}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial \phi} - \frac{\partial \mathbf{q}_i}{\partial \phi} \right) \\ \frac{\partial \mathbf{n}_i}{\partial \theta} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) + \mathbf{n}_i \cdot \left(\frac{\partial [C]}{\partial \theta}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial \theta} - \frac{\partial \mathbf{q}_i}{\partial \theta} \right) \\ \frac{\partial \mathbf{n}_i}{\partial \psi} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) + \mathbf{n}_i \cdot \left(\frac{\partial [C]}{\partial \psi}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial \psi} - \frac{\partial \mathbf{q}_i}{\partial \psi} \right) \\ \frac{\partial \mathbf{n}_i}{\partial t_x} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) + \mathbf{n}_i \cdot \left(\frac{\partial [C]}{\partial t_x}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial t_x} - \frac{\partial \mathbf{q}_i}{\partial t_x} \right) \\ \frac{\partial \mathbf{n}_i}{\partial t_y} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) + \mathbf{n}_i \cdot \left(\frac{\partial [C]}{\partial t_y}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial t_y} - \frac{\partial \mathbf{q}_i}{\partial t_y} \right) \\ \frac{\partial \mathbf{n}_i}{\partial t_z} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) + \mathbf{n}_i \cdot \left(\frac{\partial [C]}{\partial t_z}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial t_z} - \frac{\partial \mathbf{q}_i}{\partial t_z} \right) \end{pmatrix} \tag{4-15}$$

It is clear that

$$\frac{\partial \mathbf{t}}{\partial \phi} = \frac{\partial \mathbf{t}}{\partial \theta} = \frac{\partial \mathbf{t}}{\partial \psi} = 0 \quad \text{and} \quad \frac{\partial [C]}{\partial t_x} = \frac{\partial [C]}{\partial t_y} = \frac{\partial [C]}{\partial t_z} = [0] \tag{4-16}$$

By assumption, $\mathbf{q}_i$ is an orthogonal projection of $\mathbf{r}_i$ on $\mathbf{P}(u,v)$ and $\mathbf{n}_i$ is the unit normal vector to the surface $\mathbf{P}(u,v)$ at the projection $\mathbf{q}_i$. Therefore the vector $([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i)$ is collinear with the unit normal vector $\mathbf{n}_i$ and the following relationships exist:

$$\mathbf{n}_i \cdot \frac{\partial \mathbf{q}_i}{\partial \phi} = \mathbf{n}_i \cdot \frac{\partial \mathbf{q}_i}{\partial \theta} = \mathbf{n}_i \cdot \frac{\partial \mathbf{q}_i}{\partial \psi} = 0 \tag{4-17}$$

$$\mathbf{n}_i \cdot \frac{\partial \mathbf{q}_i}{\partial t_x} = \mathbf{n}_i \cdot \frac{\partial \mathbf{q}_i}{\partial t_y} = \mathbf{n}_i \cdot \frac{\partial \mathbf{q}_i}{\partial t_z} = 0 \tag{4-18}$$

$$\frac{\partial \mathbf{n}_i}{\partial \phi} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) = \frac{\partial \mathbf{n}_i}{\partial \theta} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) = \frac{\partial \mathbf{n}_i}{\partial \psi} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) = 0 \tag{4-19}$$

$$\frac{\partial \mathbf{n}_i}{\partial t_x} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) = \frac{\partial \mathbf{n}_i}{\partial t_y} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) = \frac{\partial \mathbf{n}_i}{\partial t_z} \cdot ([C]\mathbf{R}_i + \mathbf{t} - \mathbf{q}_i) = 0 \qquad (4\text{-}20)$$

Now by using the expressions of (4-16) through (4-20) the Jacobian $\boldsymbol{J}_i$ defined in (4-15) may be simplified to

$$\boldsymbol{J}_i^T = \begin{pmatrix} \mathbf{n}_i \cdot \dfrac{\partial [C]}{\partial \phi}\mathbf{R}_i \\ \mathbf{n}_i \cdot \dfrac{\partial [C]}{\partial \theta}\mathbf{R}_i \\ \mathbf{n}_i \cdot \dfrac{\partial [C]}{\partial \psi}\mathbf{R}_i \\ \dfrac{\mathbf{n}_i \cdot \partial(\mathbf{t})}{\partial t_x} \\ \dfrac{\mathbf{n}_i \cdot \partial(\mathbf{t})}{\partial t_y} \\ \dfrac{\mathbf{n}_i \cdot \partial(\mathbf{t})}{\partial t_z} \end{pmatrix} = \begin{pmatrix} \mathbf{n}_i \cdot \dfrac{\partial [C]}{\partial \phi}\mathbf{R}_i \\ \mathbf{n}_i \cdot \dfrac{\partial [C]}{\partial \theta}\mathbf{R}_i \\ \mathbf{n}_i \cdot \dfrac{\partial [C]}{\partial \psi}\mathbf{R}_i \\ n_x^i \\ n_y^i \\ n_z^i \end{pmatrix} \qquad (4\text{-}21)$$

where n_x^i, n_y^i and n_z^i are the respective scalar components of the normal vector $\mathbf{n}_i$. The rotational elements of the Jacobian require calculating the first partial derivatives of the rotational transformation matrix $[C]$ in each of the variables φ, θ and ψ. Those partial derivatives are given by the following three matrix equations.

$$\frac{\partial [C]}{\partial \phi} = \begin{pmatrix} -\cos\theta\sin\phi & -\cos\theta\cos\phi & 0 \\ \cos\psi\cos\phi - \sin\theta\sin\psi\sin\phi & -\cos\psi\sin\phi - \sin\theta\cos\phi\sin\psi & 0 \\ \sin\psi\cos\phi + \sin\theta\sin\phi\cos\psi & -\sin\psi\sin\phi + \sin\theta\cos\phi\cos\psi & 0 \end{pmatrix} \qquad (4\text{-}22)$$

$$\frac{\partial [C]}{\partial \theta} = \begin{pmatrix} -\sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \\ \cos\theta\sin\psi\cos\phi & -\cos\theta\sin\phi\sin\psi & \sin\psi\sin\theta \\ -\cos\theta\cos\phi\cos\psi & \cos\theta\sin\phi\cos\psi & -\sin\theta\cos\psi \end{pmatrix} \qquad (4\text{-}23)$$

$$\frac{\partial[C]}{\partial\psi} = \begin{pmatrix} 0 & 0 & 0 \\ -\sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi & -\sin\psi\cos\phi - \sin\theta\sin\phi\cos\psi & -\cos\psi\cos\theta \\ \cos\psi\sin\phi + \sin\theta\cos\phi\sin\psi & \cos\psi\cos\phi - \sin\theta\sin\phi\sin\psi & -\cos\theta\sin\psi \end{pmatrix} \quad (4\text{-}24)$$

The rotational elements of the Jacobian are finally determined by multiplying each of the three matrices of (4-15), (4-16) and (4-17) by $\mathbf{R}_i$ and substituting the results into (4-14).

4.4 Problem Solution

Using the constraint functions of (4-6) and (4-7) with the Jacobian of (4-14) as inputs to the optimaization routine, iterative solutions are calculated seeking a feasible set of rotations and translations to satisfy the constraints of the problem. The user provides the values for the constraints, ε and δ, which define the limits for the oriented distance function for each measured point. The appropriate constraint is determined by the proximity or non-proximity of a measured point to the leading edge of the design surface blade. If the algorithm can find a feasible solution to the problem, that solution represents a prescription for the rotations and translations which should be performed on the set of measured points to localize them and satisfy the given constraints. If a feasible solution cannot be found, then the user will need to either relax the specified constraints or remove some measured points from the set that is analyzed.

Experimental results using the constrained localization method are presented in Chapter 5.

Chapter 5

APPLICATIONS OF LOCALIZATION

5.1 Introduction

The focus of this thesis is the development of reliable computational methods for the solution of the localization problem for application in the inspection of manufactured marine propellers. A theoretical basis for these methods was presented in Chapters 3 and 4, but if the methods cannot be used for the intended application, then their theoretical development becomes only an academic exercise. The true value of the development can only be demonstrated if the methods can be used to solve **real** problems with **real** data. It is, therefore, important to show that the methods work using actual measured data and actual designs from manufactured propellers.

At the beginning of the localization investigation it was decided that measured data from a real manufactured propeller was essential to the development and validation of the localization methods. The design description of a manufactured blade and a set of measured data points from that blade were needed for testing of the localization methods as they were developed. This need was identified at a periodic meeting of the *PRAXITELES* user's group in October 1990.[7] Since it was expected that the results of the localization investigation would produce an enhancement to *PRAXITELES*, the Applied

[7] *PRAXITELES* is an interactive geometric modeling system for sculptured curves and surfaces. It has been developed in the Ocean Engineering Design Laboratory at MIT with funding from various U. S. government agencies [Hottel 91], [Tuohy 91].

Research Laboratory (ARL) at Pennsylvania State University agreed to provide design and inspection data for a real propeller [Holter 90]. This design was a fan blade that had been designed, manufactured and inspected at ARL.

At a subsequent *PRAXITELES* user's group meeting in January 1991 the preliminary results of the unconstrained localization method were presented. At this time it was suggested that a more complex test model for the localization method would be useful. While the fan blade from ARL was certainly a propeller, it did not have the complex sculptured geometry that would be typical of many marine propellers. For this reason, Philadelphia Naval Shipyard (PNSY) agreed to provide the investigator with data for another blade which would be more representative of marine propellers for localization testing [Koehler 91].

The experimental results and validation of the developed methods for unconstrained and constrained localization are presented in this chapter. Test results for the methods will be presented for both the ARL and PNSY blades. It is believed that the experimental results confirm the validity of the methods and provide a sound experimental basis for further development.

5.2 Experimental Assumptions

An essential element of the localization process is the determination of the minimum distance from a point to a surface and this distance calculation is related to the orthogonal projection of the point onto the surface. The orthogonal projections can be readily determined by exploiting some conditions that are assumed to exist in the normal inspection of manufactured propellers:

(1) The measured points on the manufactured surface can be expected to lie sufficiently near to the design surface so that the minimum distance projection point onto the design surface is unique.[8]

(2) The measurements are made away from the tip, root edge or trailing edge of the blade, so the minimum distance projection point is also an orthogonal projection of the measured point onto the surface.

(3) The inspection device makes measurements by moving in an organized manner over the blade. The measurement might be made in linear or curved passes, but it is not random in its acquisition of data.[9]

If a cubic B-spline curve is fitted through a set of measured points using a least squares fitting routine with chord length parametrization, then the orthogonal projection of each fitted curve will provide a mapping of u,v parameters for each measured point in the parametric space of the design surface. This mapping approximates the orthogonal projection of each measured point having parameter t with the orthogonal projection of the point on the least squares fitted curve having the same parameter t. Using these parameters as a starting point, the orthogonal projection routine for a particular measured point will converge rapidly to the parameters in the design surface which correspond to the actual orthogonal projection. The method works because the measured points are near the design surface for the manufactured surface under consideration, and because

---

[8] Further analysis of this assumption can be found in [Patrikalakis 90], [Kriezis 90], [Kriezis 91], and [Wolter 85].

[9] This is not an essential assumption or a particularly strong one. The methods of measurement for automated inspection normally involve numerical programming of the inspection device that is analogous to the numerical programming of numerically controlled machinery. The assumption of organized paths allows for straightforward correlation of sets of measured data points and this correlation increases the efficiency of the localization process.

there are no discontinuities in the region of measurement. The method will *not* work at points of discontinuity or at points which are very near to the cut loci of the blade [Wolter 85].[10]

5.3 Applied Research Laboratory Propeller

The propeller design that was provided by ARL was received as a NURBS surface. The inspection data was received as x, y, z coordinates of measured points on the blade that was manufactured from the NURBS design description. These measurements were made at ARL using the Intelligent Robotic Inpection System (IRIS), which uses laser interferometry to obtain highly accurate measurements of surface coordinates. 1381 data points were received representing measurements on the pressure and suction sides of the blade. The design surface was a bicubic NURBS patch parameterized with 53 knots in the u direction and 20 knots in the v direction. The blade had a nominal radius of 12 inches from root to tip.

Some minor preprocessing was required to get the measured data into a form that was suitable for localization with the developed methods. The first step in this preprocessing was to visually inspect the data received from ARL to establish a method for correlating the inspection data points. This was accomplished by using the visualization capabilities that exist in *PRAXITELES*. The blade and measured points from ARL had the appearance shown in Figure 1.

---

[10] These unsatisfactory conditions typically exist at the trailing edge or tip of the blade, where the minimum distance projection of a point onto the surface of the blade is not uniquely defined. This fact is the basis for assumptions (1) and (2) above.

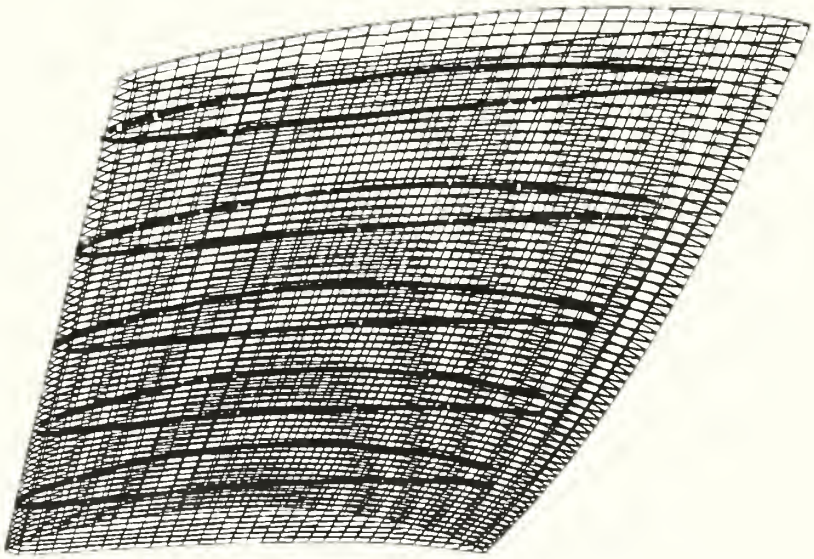

Figure 1. ARL Propeller Blade Showing Measured Points

It was clear that the inspection measurements at ARL formed 10 correlated sets of points. These 10 sets of points were segregated into 10 groups of points through which a cubic B-spline curve was fit using a least squares approximation with chord length

parametrization and zero internal knots.[11] The orthogonal projection of the cubic B-spline curve onto the design surface was used as the source of u,v parameters for initialization of the orthogonal projection routines for individual measured points. With this initialization the orthogonal projection routines converge rapidly to the u,v parameters in the design surface for the actual orthogonal projection of each measured point. The u,v parameters of the projection are identical to the u,v parameters in the design surface at the point of minimum distance from the measured point. Using this information it is possible to establish a baseline file which contains the original measured points, the u,v parameters in the design surface at the points of minimum distance, and finally the minimum distances themselves. The distances at each point in the file form an initial reference condition to which the results of localization can be compared.

5.3.1 Unconstrained Localization Results

Further processing of the file containing the parametrizations of the minimum distance points in the design surface was performed to ensure that experimental assumption (2) was satisfied, i.e. to ensure that points in the interior of the blade were well away from the trailing edge, root edge or tip of the blade. Visual inspection in *PRAXITELES* revealed that chordwise measurements of the blade corresponded closely with the u parameter of the design surface, and predictably the span of the blade corresponded closely with the v parameter in the design surface. This condition made it possible to remove points from the total set of measured points which might

[11] This produces the simplest cubic B-spline, a Bezier curve. It is not necessary for the method to use any more knots or higher order B-spline than this. Using a Bezier curve provides for rapid fitting of the measured data points with sufficient accuracy to quickly find an orthogonal projection from each point to the design surface.

be too close to the tip or trailing edge of the blade. This was accomplished by choosing points whose projections in the design surface had u,v parameters which satisfied the condition ($0.05 \leq u,v \leq 0.95$). This resulted in an initial data file for localization containing 1261 points, which was 8.7 per cent smaller than the original set of data points.

$$\text{Root Mean Square Distance} = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \tag{5-1}$$

The results of the unconstrained localization and the computational time required for each set of points is presented in Table I.[12] The translations and rotations for each set of points after unconstrained localization are presented in Table II.

Table I

Number of Points	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computation Time (CPU)
21	0.02371	0.01866	-21.3	0 min 40 sec
41	0.03196	0.02593	-18.9	1 min 25 sec
61	0.03016	0.02435	-19.3	1 min 56 sec
85	0.03130	0.02485	-20.6	2 min 45 sec
106	0.03010	0.02447	-18.7	4 min 9 sec

RMS Distances and Computation Times for Unconstrained Localization of ARL Propeller

[12] The computations for the experimental results of this thesis were performed on a Silicon Graphics 4D25TG "Personal Iris" machine running at a nominal speed of 1.6 million floating point operations per second.

Table II

Number of Points	Translation (inches)			Rotation (radians)		
	t_x	t_y	t_z	φ	θ	ψ
21	0.1796	-0.0215	-0.2896	0.0005	-0.0058	-0.0005
41	0.0887	-0.0627	-0.1069	0.0041	0.0005	-0.0023
61	0.1018	-0.0529	0.0538	0.0025	-0.0073	-0.0023
85	0.1094	-0.0645	0.0914	0.0034	-0.0077	-0.0028
106	0.0827	-0.0501	0.0135	0.0025	-0.0047	-0.0020

Translations and Rotations for Unconstrained Localization of ARL Propeller

The transformation produced for the unconstrained localization of the ARL propeller produced an average reduction in root mean square distance from the sampled point set to the design surface of 19.8 per cent. The computational time required for the entire set of 1261 points was 1 hour and 7 minutes, producing a reduction in root mean square distance of 19.6 per cent.[13]

To test the validity of the transformation obtained using the unconstrained localization method, the inverse of the transformation for the 21 point sample was applied to the original design surface. When the same points were localized to the *transformed* design surface the new transformation was the zero vector. *PRAXITELES* was then used to obtain a set of points on the transformed design

[13] The computational time and reduction of root mean square distance for 1261 points is listed as a benchmark for the the performance of the unconstrained localization method on a very large data set. Although the results are consistent with those presented in Table I, they should be considered only as a measure of the time required for a very large data set. Other examples for very large data sets were not tested because of the long time required, and because data sets larger than about 100 data points did not seem to have a statistically significant effect upon the results of the localization process.

surface. When these points were localized relative to the *original* design surface, the same transformation was produced as that using the 21 point sample of measured points. These results provide experimental confirmation of the Euclidean property of the localization transformation, as discussed earlier in Section 3.3 and shown in Appendix A.

5.3.2 Constrained Localization Results

The constrained localization problem differs from the unconstrained localization problem because measured points near the leading edge of a manufactured blade have greater influence on the localization than do points in other parts of the blade. The method uses the unconstrained localization as a starting point with the implicit assumption that global minimization of distances of measured points to the design surface is achieved before the start of the constrained localization. The constrained localization algorithm also uses the oriented distance function as a distance measure rather than the squared distance function. This distance function provides a highly accurate estimate of distance and is fundamental to the constrained localization method as developed in Chapter 4.

The constrained localization algorithm was evaluated using the same datasets that were used for the unconstrained localization problem. The **leading edge** region of the blade was defined as the set of u,v parameters of the design surface where the condition ($0.4 \leq u \leq 0.6$) existed. The **non-leading edge** region of the blade was defined as the set of u,v parameters for which this condition did not occur. This selection was based upon visual observation in *PRAXITELES* of the ARL blade. The leading edge of the blade was almost exactly coincident with the isoparameter line

$u = 0.5$ in the design surface of the blade. With this definition of the leading edge region, the measured points are uniquely mapped to either a *leading edge region* or a *non-leading edge region* based upon the value of the parameter u for the projection of each point onto the design surface at the beginning of the constrained localization process. Clearly the set of points in the leading edge region and the set of points in the non-leading edge region are complementary subsets of the universal set of measured points.

Since the testing of the constrained localization method was intended primarily to demonstrate the viability of the method, the absolute magnitudes of the constraints were not considered as important as the relative magnitudes of the constraints for each region. Using this philosophy, the constraints were arbitrarily assigned with relative magnitudes having a 10 to 1 ratio. This means that the distance constraint in the non-leading edge region of the blade was assigned a magnitude of 10 times the magnitude of the distance constraint in the leading edge region. This constraint assignment strategy and the conditions necessary for a satisfactory constrained localization can be summarized in the following statements.

(1) The constraints in the leading edge region of the blade have one-tenth the magnitude of the constraints in the non-leading edge region of the blade.

(2) The condition for satisfactory constrained localization requires two necessary corollary conditions:

a) All measured points that are assigned to the leading edge region of the blade must have post-localization minimum distances to the design surface which are less than or equal to the magnitude of the leading edge constraint.

b) All measured points that are assigned to the non-leading edge region of the blade must have post-localization minimum distances to the design surface which are less than or equal to the magnitude of the non-leading edge constraint.

The procedure that was used to evaluate the constrained localization algorithm involved selecting a value for the leading edge and non-leading edge constraints below the threshold where a feasible localization solution could be obtained for a given set of measured data points. Observing that the fixed ratio between the magnitudes of the constraints was always maintained, the values of the constraints were incrementally increased until a feasible constrained localization solution could be found. This experimental procedure thereby determined a lower bound on the values of constraints which could produce feasible solutions to the constrained localization problem for the given set of measured points.[14] The experimental results for the constrained localization of points near the leading edge and non-leading edge regions of the ARL blade are presented in separate tables for clarity. The root mean square distances, computed using equation (5-1), and the computation times for the two regions are presented in Tables III and IV.[15] The corresponding translations, rotations and maximum distances are presented in Table V.

---

[14] In a manufacturing setting the magnitudes of the constraints would be specified by the blade designer. Those magnitudes might not have the same fixed ratio that was used in these experiments.

[15] The computation times shown in Tables III and IV are the same because the *same* sets of points were used for each table. They are duplicated in the two tables for easy reference.

Table III

Number of Points	Leading Edge Constraint	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computational Time (CPU)
21	0.012 in	0.01408	0.01014	-28.0	0 min 7 sec
41	0.012 in	0.02038	0.00888	-56.4	0 min 13 sec
61	0.017 in	0.01552	0.01290	-19.8	0 min 36 sec
85	0.017 in	0.01542	0.01168	-24.2	0 min 27 sec
106	0.017 in	0.01529	0.01046	-50.0	1 min 8 sec

RMS Distances and Computation Times
for Constrained Localization of ARL Propeller (Leading Edge)

Table IV

Number of Points	Non-Leading Edge Constraint	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computational Time (CPU)
21	0.12 in	0.02201	0.04251	+134.0	0 min 7 sec
41	0.12 in	0.02749	0.04241	+54.3	0 min 13 sec
61	0.17 in	0.02642	0.04877	+185.6	0 min 36 sec
85	0.17 in	0.02697	0.03902	+44.7	0 min 27 sec
106	0.17 in	0.02657	0.03935	+35.0	1 min 8 sec

RMS Distances and Computation Times
for Constrained Localization of ARL Propeller (Non-Leading Edge)

Table V

Number of Points	Translation (inches)			Rotation (radians)			Leading Edge Max Distance	Non-Leading Edge Max Distance
	t_x	t_y	t_z	φ	θ	ψ		
21	0.1712	-0.0866	0.5881	0.0043	-0.0045	-0.0043	0.01200 in	0.11609 in
41	0.0821	-0.0324	-0.0966	0.0004	-.0096	-0.0011	0.01200 in	0.12000 in
61	-0.0018	0.0077	0.1717	-0.0038	-0.0054	-0.0008	0.01700 in	0.11643 in
85	-0.1185	-0.0887	0.6217	0.0040	-0.0084	-0.0051	0.01700 in	0.11593 in
106	-0.1788	-0.0629	0.6799	0.0034	-0.0062	-0.0036	0.01700 in	0.12961 in

Translations, Rotations and Maximum Distances for Constrained Localization of ARL Propeller

It should be noted that the translation t_z in Table V is typically much larger than the translations t_x and t_y. This result is due to the particular orientation of the blade relative to the axes of the coordinate system in which the measurements are made. The *z-axis* in this example is nearly parallel to the span of the blade, so that points at extreme positions of the *z-axis* must be near to the root edge and tip edge of the blade. These are the points which limit translational motion along the *z-axis* [Gunnarsson 87a]. Since points very close to these edges are excluded from the data set at the beginning of the localization, there are necessarily relatively few to constrain motion in the z direction compared with the number of points that constrain motion in the x and y directions.

The transformation produced for the constrained localization of the ARL propeller reduced the root mean square distance from the sampled point set to the design surface near the leading edge by an average of 35.7 per cent and increased the

root mean square distance from the sampled point set to the design surface away from the leading edge by an average of 90.7 per cent. The increase is a direct result of reducing the root mean square distance near the leading edge and transforming the point set away from a condition of global minimization. All points of each sampled point set had an absolute distance which was less than or equal to the listed constraint. The effects of the constrained localization on the global root mean square distances are presented in Table VI.

Table VI

Number of Points	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change
21	0.01866 in	0.03155 in	+69.1
41	0.02593 in	0.03825 in	+47.4
61	0.02435 in	0.04325 in	+72.7
85	0.02485 in	0.03483 in	+38.4
106	0.02447 in	0.03535 in	+41.9

Global Localization Effects
for Constrained Localization of ARL Propeller

The global root mean square distance for the ARL propeller increased an average of 53.9 per cent for the five sampled point sets after the constrained localization process.

5.4 Philadelphia Naval Shipyard Propeller

The propeller information that was received from Philadelphia Naval Shipyard was considerably more difficult to analyze than the data that was received from ARL because PNSY could not provide an analytic design description of the propeller blade in the form of a NURBS surface. Instead PNSY provided two sets of *measurements* of a manufactured blade. This created significant problems because the localization methods require a NURBS surface description of the design surface for their proper operation.

The measured data from PNSY was obtained using the Automated Propeller Optical Measurement System (APOMS). It uses a laser interferometry technique similar to that of the IRIS system at ARL. The measurements were made on a submarine propeller blade that was manufactured many years ago by Philadelphia Naval Shipyard.[16] The blade had an approximate radius of 68 inches form root to tip, and was therefore about 6 times larger than the blade received from ARL.

In order to properly test the methods of unconstrained and constrained localization, it was necessary to obtain a "design surface" from the set of measured points. This was accomplished by least squares fitting of cubic B-spline curves through the measured data points, followed by lofting a bicubic B-spline surface through the set of B-spline curves

[16] The measured data for this blade was certified by Commander, Philadelphia Naval Shipyard and Naval Sea Systems Command (Code 56X73) to be not of a classifiable nature in PNSY letter 9245, Code 266, Serial 9166002 of 6 February 1991.

so generated. Chord length parametrization was used for the curves with sixty knots.[17] The surface which was produced could be used as a "design surface" for localization testing with the implicit understanding that the surface is not the *actual* design surface as in the case of the ARL blade, nor is it necessarily the best possible surface representation of the measured data. The lofted surface is not identical to the surface from which the blade was originally manufactured; it is merely an approximation of that surface. For this reason the lofted surface will introduce some error into the localization process and will ultimately reduce the level of accuracy that can be achieved from it.

The use of a lofted surface produced from measured data at PNSY as a "design surface" arose out of necessity. It was not possible to obtain another blade with both a NURBS surface description of the underlying design and with inspection data of the manufactured blade as well. For this reason the PNSY design surface that was generated through lofting must be viewed as a **simulated** design surface. This example is not as good as the ARL example because the original design surface was not available for the corresponding inspection data.[18] On the other hand, the example illustrates the usefulness of the localization methods quite adequately.

---

[17] A uniform spacing scheme was used for the distribution of internal knots and for ease in lofting. The curve is *not* the very best approximation of the data; significant error arises near the trailing edge region of the blade because of the sparcity of knots. Non-uniform knots with a higher concentration of knots near the trailing edge would improve the approximation for individual curves. However, this scheme would greatly increase the size and complexity of the lofted surface and it was not deemed necessary for the demonstration of these experiments.

[18] It should be noted that marine propeller design surfaces typically arise from lofting a surface through a set of curves which describe a hydrofoil section. The surface so produced is *defined* as the "design surface". Therefore the lofted surface of this example is a "design surface". The important distinction is that this design surface is not the *same* design surface as the one used for the manufacture of the propeller.

Despite the previously mentioned problems, the set of "design" points that was received from Philadelphia allowed a reasonably good blade surface to be generated. The "design" points were received in 17 sets of cylindrical measurements taken in radial cuts from the root to the tip of the blade. Each set of measurements contained 180 points distributed around the surface of the blade. The B-spline curves generated from these points and the bicubic NURBS surface that was lofted through them were all produced using *PRAXITELES*.[19]

The other set of measured points that was received from Philadelphia had similar form to the 17 sets of points that were used to produce the design surface. There were 18 radial cuts taken on the pressure and suction sides of the blade at radii extending from 0.3 to 0.99. The points at radii above 0.90 are very close to the tip of the blade, so to ensure that all points used were relatively far from the blade tip, only points taken at radii from 0.3 to 0.8 were used. These points and the lofted surface were visually inspected using *PRAXITELES*. The "design" blade surface and measured points for this blade had the appearance shown in Figure 2.[20]

---

[19] The bicubic NURBS surface was generated using uniformly spaced internal knots with a total of 17 knots in the u direction and 56 knots in the v direction. The selection of direction for the u,v axes used for lofting was arbitrary. The resulting "design surface" parametric axes are orthogonal to the ARL design surface parametric axes by pure coincidence.

[20] As in the case of the unconstrained localization of the ARL blade the 12 "curves" of measured data which are visible in Figure 2 were fitted with cubic Bezier curves to facilitate finding the orthogonal projection of each measured point onto the design surface.

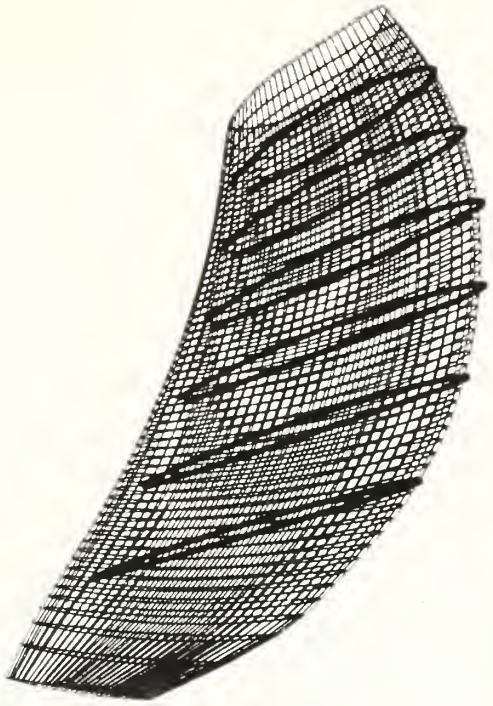

Figure 2. PNSY Propeller Blade Showing Measured Points

5.4.1 Unconstrained Localization Results

A total of 4077 measured *inspection* points from the manufactured blade were received from Philadelphia. To prevent problems with points near parametric boundaries of the design surface, these points were culled to a set of 3214 points by imposing the same condition that was used for the ARL blade. Points were chosen whose projections in the design surface had u,v parameters which satisfied the condition ($0.05 \leq u, v \leq 0.95$). This smaller set of points represented a 21 per cent reduction in the number of measured points.

Subsets of the global set of 3214 points were extracted for testing using the same procedures as those used for the ARL blade. Points were pseudorandomly selected over the pressure and suction sides to produce five sets of measured points for evaluation. The points were distributed over the entire surface of the blade to provide a good distribution of data. The root mean square distance, calculated using (5-1), was again used as a measure of the performance of the unconstrained localization algorithm. The results of the evaluation for five subsets of points are presented in Tables VII and VIII.

Table VII

Number of Points	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computation Time (CPU)
21	0.15599	0.11430	-26.7	1 min 14 sec
41	0.15798	0.09763	-38.2	2 min 51 sec
61	0.16103	0.11528	-28.4	2 min 20 sec
81	0.16291	0.10698	-34.3	2 min 44 sec
101	0.14236	0.10966	-23.0	3 min 20 sec

RMS Distances and Computation Times
for Unconstrained Localization of PNSY Propeller

Table VIII

Number of Points	Translation (inches)			Rotation (radians)		
	t_x	t_y	t_z	φ	θ	ψ
21	0.1278	-0.0622	-0.1460	-0.0025	-0.0011	-0.0006
41	0.3079	-0.2866	-0.2232	0.0043	0.0029	0.0017
61	-0.0682	-0.4959	-0.0666	-0.0078	0.0028	-0.0011
81	0.1580	-0.5114	-0.1442	-0.0022	0.0035	0.0009
101	-0.0097	-0.4866	-0.0736	-0.0061	0.0030	-0.0006

Translations and Rotations
for Unconstrained Localization of PNSY Propeller

The unconstrained localization of the five sets of measured points reduced the root mean square distance from the points to the design surface by an average of 30.1 per cent.

5.4.2 Constrained Localization Results

The constrained localization testing of the PNSY propeller used the same sets of data and essentially the same procedures as those that were used for the ARL blade. Points were assigned to a *leading edge* region or a *non-leading edge* region of the blade based upon the position of the projection of the point onto the design surface at the beginning of the constrained localization procedure. A procedural distinction between the PNSY blade and the ARL blade was the direction of the u,v parameters in the design surface. While the leading edge for the ARL blade was nearly coincident with the isoparameter line $u = 0.5$, the PNSY blade had the leading edge nearly coincident with the isoparameter line $v = 0.5$. This difference required a change in the definition of the leading edge and non-leading edge regions of the blade. The **leading edge** region of the blade was defined by the condition $0.4 \leq v \leq 0.6$. The **non-leading edge** region of the blade was the region where this condition did not occur. Using these definitions a given set of measured points was mapped to specific regions of the design surface as was done with the ARL blade.

The same assignment of the ratio of constraint magnitudes was used as in the ARL blade testing. The points assigned to a leading edge region had a constraint with magnitude equal to one-tenth of the magnitude of the constraint for points in the non-leading edge region of the blade. For testing of the process, the constraint for

each region was incrementally increased until a threshold was reached where a feasible solution to the constrained problem was obtained. In this way, a lower bound was determined for constraints which could provide feasible solutions to the problem for each set of points. The experimental results using this procedure are presented in Tables IX through XI.[21]

Table IX

Number of Points	Leading Edge Constraint	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computational Time (CPU)
21	0.11 in	0.13280	0.09419	-29.1	0 min 12 sec
41	0.12 in	0.11777	0.09560	-18.8	0 min 50 sec
61	0.12 in	0.10991	0.07713	-29.8	0 min 22 sec
81	0.15 in	0.12298	0.10382	-15.6	0 min 18 sec
101	0.13 in	0.11247	0.09010	-19.9	1 min 22 sec

RMS Distances and Computation Times
for Constrained Localization of PNSY Propeller (Leading Edge)

Table X

Number of Points	Non-Leading Edge Constraint	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computational Time (CPU)
21	1.1 in	0.10382	0.25875	+149.2	0 min 12 sec
41	1.2 in	0.08793	0.21682	+124.1	0 min 50 sec
61	1.2 in	0.11728	0.31872	+171.8	0 min 22 sec
81	1.5 in	0.09993	0.20267	+102.8	0 min 18 sec
101	1.3 in	0.10862	0.27141	+149.9	1 min 22 sec

RMS Distances and Computation Times
for Constrained Localization of PNSY Propeller (Non-Leading Edge)

[21] The computational times shown in Tables IX and X are the same because the tests were performed at the same time using the same set of points. The times are duplicated for easy reference.

Table XI

Number of Points	Translation (inches)			Rotation (radians)			Leading Edge Max Distance	Non-Leading Edge Max Distance
	t_x	t_y	t_z	φ	θ	ψ		
21	0.1401	-0.0495	-0.1540	-0.0035	-0.0145	-0.0006	0.11000 in	0.48742 in
41	0.3222	-0.2930	-0.2317	0.0015	-0.0097	0.0021	0.12000 in	0.48649 in
61	-0.3292	0.2712	0.0487	-0.0160	-0.0196	-0.0065	0.11119 in	0.67443 in
81	0.1762	-0.4528	-0.2197	-0.0013	-0.0080	0.0021	0.14612 in	0.45780 in
101	-0.1731	0.2832	-0.1185	-0.0122	-0.0160	-0.0032	0.12856 in	0.60269 in

Translations, Rotations and Maximum Distances for Constrained Localization of PNSY Propeller

The transformation produced for the constrained localization of the PNSY propeller reduced the root mean square distance from the sampled point set to the design surface near the leading edge by an average of 21.2 per cent. All points of each sampled point set had an absolute distance which was less than or equal to the listed constraint. The transformation produced for the constrained localization of the PNSY propeller increased the root mean square distance from the sampled point set to the design surface away from the leading edge by an average of 139.6 per cent. This increase was the result of reducing the root mean square distance near the leading edge and transforming the point set away from a condition of global minimization. The effects of the constrained localization on the global root mean square distances are presented in Table XII.

Table XII

Number of Points	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change
21	0.11430	0.21816	+90.8
41	0.09762	0.18954	+94.2
61	0.11528	0.27373	+137.4
81	0.10698	0.18020	+68.4
101	0.10966	0.23703	+116.1

Global Localization Effects
for Constrained Localization of PNSY Propeller

The global root mean square distance for the ARL propeller increased an average of 101.4 per cent for the five sampled point sets after the constrained localization process. The entire set of 3214 points was not tested.

The results of this test showed that the lower bound on the constraint generally increased with the number of points tested, but not in all cases. The constraint for 81 points was higher than that for 101 points. An important consideration is that the results represent only a *feasible* solution to the problem for a given set of points. This solution is not intended to be a global minimization of the root mean square distance, nor is it likely to be unique.

5.5 Unconstrained Localization of Multiple Surface Patches

A current practice in the design of marine propellers is to break the blade surface up into several regions and to provide a separate design surface patch description for each individual region, rather than providing a single surface description of the entire blade. The blade may be broken up into separate patches for the leading edge, for the trailing

edge, for the pressure side, for the suction side, for hub fillets, etc. Each of these regions might in fact be broken into separate surfaces as well. It is, therefore, important that a localization method for marine propellers be able to accomodate the multiple surface patches that might exist in a real propeller design description.

The unconstrained localization algorithm presented in Chapter 3 was generalized to accomodate multiple untrimmed NURBS patches. The two design surfaces and measured points of the ARL and PNSY propellers were used to test the algorithm. Each design surface was divided into three patches to represent the leading edge, pressure side and suction side regions of a multiple surface blade. These three patches were used to simulate the design representation of a multiple patch blade. It is assumed that all patches of such a blade are untrimmed NURBS patches of orders M in u, and N in v, $(0 \leq u, v \leq 1)$ with at least tangent plane continuity.

5.5.1 ARL Propeller Blade

PRAXITELES was used to split the ARL blade into NURBS patches comprising approximately one-third of the original design surface each. The surface was split along the two isoparameter lines $u = 0.33$ and $u = 0.66$, and each patch was reparameterized such that $(0 \leq u, v \leq 1)$. Points were then assigned to the three patches based upon the u,v parameters of the projection of each point onto the original design surface. To allow for problems that might arise for points very near to parametric boundaries, a "buffer" region of parametric values was assigned at the boundaries of the three patches. This buffer region was set equal to 0.02 in each

parameter, so that each patch was reduced by this amount around its edges. Using these ideas the original design surface and the 1261 original measured points produced three untrimmed patches and point sets having the following characteristics.

Pressure Side Patch ($0.02 \leq u \leq 0.31$ and $0.02 \leq v \leq 0.98$) 402 points

Leading Edge Patch ($0.35 \leq u \leq 0.64$ and $0.02 \leq v \leq 0.98$) 394 points

Suction Side Patch ($0.68 \leq u \leq 0.98$ and $0.02 \leq v \leq 0.98$) 424 points

The points assigned to the three regions of the blade were pseudorandomly selected to produce five sets of pseudorandom data as used in previous testing. (These point sets had comparable numbers of points, but were not the same point sets.) The root mean square distances from the sets of points to the surface patches were calculated using (5-1). The results of the unconstrained localization of these sets of data relative to the three surface patches are presented in Tables XIII and XIV.

Table XIII

Number of Points	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computation Time (CPU)
20	0.03406	0.02574	-24.4	0 min 37 sec
41	0.03399	0.02416	-28.9	1 min 34 sec
61	0.03269	0.02517	-23.0	2 min 3 sec
82	0.03346	0.02524	-24.6	2 min 31 sec
102	0.03317	0.02576	-22.3	3 min 31 sec

RMS Distances and Computation Times
for Unconstrained Localization of Three Patches from ARL Propeller

Table XIV

Number of Points	Translation (inches)			Rotation (radians)		
	t_x	t_y	t_z	φ	θ	ψ
20	0.2275	-0.0673	-0.2386	0.0011	-0.0080	-0.0032
41	0.2090	-0.0801	-0.1102	0.0031	-0.0078	-0.0035
61	0.1018	-0.0587	0.1361	0.0035	-0.0005	-0.0022
82	0.1440	-0.0616	-0.0585	0.0029	-0.0063	-0.0026
102	0.1279	-0.0638	-0.0722	0.0031	-0.0043	-0.0026

Translations and Rotations for Unconstrained Localization of Three Patches from ARL Propeller

Using the three patches from the ARL blade the unconstrained localization algorithm produced a transformation which reduced the root mean square distance from the points to the surface patches by an average of 24.6 per cent. The unconstrained localization for a single patch which described the entire surface produced an average reduction in root mean square distance of 19.8 per cent using different sets of measured points.

5.5.2 PNSY Propeller Blade

Testing of multiple patches from the Philadelphia blade followed a scheme virtually the same as that used for the ARL blade. The surface was split using *PRAXITELES*, but since the blade was oriented with the leading edge at $v = 0.5$ in the parametric space of the design surface, it was necessary to define the patches in terms

of the parameter v rather than u. The 3214 points of the PNSY design surface were assigned analogously to those in the ARL tests with the patches and point sets having the following characteristics.

Pressure Side Patch $(0.02 \leq u \leq 0.98$ and $0.02 \leq v \leq 0.31)$ 857 points

Leading Edge Patch $(0.02 \leq u \leq 0.98$ and $0.35 \leq v \leq 0.64)$ 1146 points

Suction Side Patch $(0.02 \leq u \leq 0.98$ and $0.68 \leq v \leq 0.98)$ 939 points

When subsets of data points were once again generated from these sets of points, the unconstrained localization algorithm was tested for the three patches of the PNSY blade using exactly the same procedure as that which was used for the ARL blade. The results of this testing are presented in Tables XV and XVI.

Table XV

Number of Points	RMS Distance Before Localization (inches)	RMS Distance After Localization (inches)	Per Cent Change	Computation Time (CPU)
21	0.19794	0.13681	-30.9	0 min 37 sec
41	0.17408	0.10136	-41.8	3 min 10 sec
61	0.16325	0.11439	-29.9	1 min 57 sec
82	0.15866	0.09463	-40.4	2 min 54 sec
102	0.15417	0.11440	-25.8	3 min 33 sec

RMS Distances and Computation Times for Unconstrained Localization of Three Patches from PNSY Propeller

Table XVI

Number of Points	Translation (inches)			Rotation (radians)		
	t_x	t_y	t_z	φ	θ	ψ
20	-0.3495	-0 3568	-0.0381	-0.0149	-0.0003	-0.0029
41	0.2840	-0.4473	-0.1510	0.0010	-0.0024	-0.0011
61	0.1354	-0.6949	-0.1327	-0.0024	0.0035	0.0011
82	0.3346	-0.9238	-0.1712	0.0030	0.0063	0.0033
102	0.1685	-0.2034	-0.1401	-0.0012	0.0000	0.0000

Translations and Rotations
for Unconstrained Localization of Three Patches from PNSY Propeller

The localization algorithm produced a transformation which reduced the root mean square distance from the points to the surface patches by an average of 33.8 per cent. This compares with an average reduction of 30.1 per cent for the single patch description of the blade with different sets of measured points.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary of Results of Investigation

This thesis has presented the theoretical development and numerical implementation in efficient computer codes of the unconstrained and constrained localization algorithms for application to the automated inspection of marine propellers. Experimental results using actual marine propeller designs with measured data have been provided to demonstrate the validity of the method. In each example, it has been shown that by using the rigid body transformation provided by the localization method a set of measured inspection points can be brought into closer agreement with a design surface. The unconstrained localization algorithm provides a method for minimizing the sum of squares of minimum distances from the measured points to the surface. The constrained localization algorithm provides a method for satisfying localized constraints, so that points near particular regions of the design surface are brought closer to the surface than other points near other regions of the surface. The latter method provides special usefulness for the problem of localizing inspection points near the leading edge of a marine propeller blade.

6.2 Projected Benefits of Investigation

It is presumed that the products of this investigation, in the form of the unconstrained and constrained localization methods with associated computer codes, will

find direct application in the inspection of manufactured marine propellers. The methods are flexible in their requirements for input information, requiring only a NURBS description of a blade and corresponding spatial coordinates of measured points on a manufactured surface. These pieces of information can be readily obtained from existing systems in the industry. The unconstrained localization method is expected to be used in an investigation by Bird-Johnson Company of Walpole, Massachusetts and David Taylor Research Center in Carderock, Maryland under a U. S. Navy contract.[22] The method would be used to evaluate the inspection results of an actual marine propeller blade. Westinghouse Machinery Technology Division of Pittsburgh, Pennsylvania, the Applied Research Laboratory at Pennsylvania State University, Metal Working Technology Corporation of Johnstown, Pennsylvania and Martin Marietta Energy Systems at Oak Ridge National Laboratory have all expressed interest in the process for future marine propeller inspection applications. These applications are precisely ones which were anticipated during the development of the method.

Beyond the direct application of the localization method which has already been implemented, it is expected that the development will prove useful in the area of better programming of work during propeller blade manufacturing. Specifically, it will be possible to better evaluate initial blade castings by confidently determining if an initial casting satisfies dimensional requirements. It is expected that fewer castings will be wasted because the manufacturing engineer will be able to determine the proper orientation of a casting to "find the blade" in a casting which might have otherwise been rejected. It seems reasonable that the localization methods will aid in planning and

[22] A preliminary users manual was produced directly from the work of this thesis to aid designers and manufacturing engineers in using the available tools [Jinkerson 91].

execution of postcasting work as well. In particular, if a blade is placed in the localized position before it is attached to the hub, and if the attachment flange is made to conform to the localized blade, then an improvement in the conformance of the overall propeller to the original design can be expected. Finally, the localization method should be very valuable in the inspection of a blade before final acceptance. If the transformation returned from the localization process is smaller than specified tolerances for a specified set of measured points, then the blade could be considered to satisfy the inspection criteria for acceptance.

6.3 Areas for Further Investigation

Perhaps the one area of this work which shows the most need for further investigation is the existing problem of finding a suitable orthogonal projection of measured points at the boundaries of the design surface. In Chapter 5 the measured data points were carefully selected to ensure that their orthogonal projections would fall well within the parametric boundaries of the design surface. The localization methods, as they presently exist, will fail if this condition is not satisfied. Such a situation needs to be corrected. Extensive preprocessing of data is needed to ensure that the localization will work for points near parametric boundaries. Further development is needed to provide a value for the point projections at the boundaries.

In Chapter 4 an assumption in the development of the constrained localization problem was that the projection of a measured point on the design surface would not change during the localization process. Although this assumption is almost certainly valid for very small rotations and translations, it constrains the flexibility and generality of the process. Rather than fixing the mapping of measured points to assigned regions

from beginning to end of the process, it would be desirable for the assignment to be changeable as the projection of a point may change during the localization. This problem is not expected to be trivial. It is expected that discontinuities will result as constraints change dynamically, and these may be difficult to handle with current optimization codes. More theoretical development will probably be necessary to address this problem. Alternatively, a smooth bivariate function might be employed to represent the constraints throughout the patch, rather than by using the piecewise constant constraints that were used in this work.

The constrained localization problem in its present formulation produces a feasible but not necessarily unique solution. It starts from a position of presumed global minimization, which is hopefully achieved during the unconstrained localization process, and seeks a condition which will satisfy local constraints. A much more difficult problem is one which would optimize the solution to the constrained localization problem. It is clear from the premise that the constrained problem starts from a global minimum, that any subsequent solution cannot be an unconstrained global minimum. Such a solution will however satisfy the optimality condition of *constrained* global minimization. A related problem would be to show that the unconstrained localization does in fact produce a global minimum.

Another area of investigation that would be worthwhile involves the application of the localization methods, constrained *and* unconstrained to the problem of trimmed NURBS patches. Untrimmed patches have been addressed in this work, but trimmed patches will be more difficult. As a minimum, a satisfactory solution to the problem of finding the orthogonal projection of a measured point at the boundaries of a design surface must be found. Solution of the trimmed patch problem will, however, greatly

increase the generality of the localization methods.

The accuracy and efficiency of the constrained localization algorithm presented in Chapter 4 was improved by providing symbolic Jacobians for the constraint function and objective function. It is likely that the unconstrained localization algorithm could be improved by supplying the Jacobian or higher order derivatives to the minimization routine as well. The Jacobian for the squared distance function is derived in Appendix B. An obvious next step in the future development of the localization process is to implement it in the unconstrained localization algortihm.

A final area that is worthy of investigation involves the application of statistical theory to the selection of measured points for evaluation and to the results produced from the methods. While the simple root mean square distance measure is an appropriate one for macroscopic evaluation of the results of the method, it would be very interesting to apply some statistical measures to the results in order to evaluate the specific effects of points in various regions of the blade. Furthermore, the selection of numbers of measured points to be used in the evaluations of the examples of Chapter 5 was consistent but arbitrary. Statistical experiments should be performed to determine the appropriate size of point sets for a given set of measured data.

Appendix A

DEMONSTRATION OF THE ORTHOGONALITY OF THE ROTATIONAL TRANSFORMATION MATRIX

A 3 by 3 matrix $[A]$ is defined to be **orthogonal** if it can be shown that

$$[A][A]^T = [A]^T[A] = I_3 \tag{A-1}$$

where I_3 is the **identity** matrix of dimension 3.

It was asserted in Chapter 3 that the **rotational transformation matrix** presented there was orthogonal. It will be shown through a direct, albeit somewhat tedious, application of matrix algebra that equation (A-1) is satisfied and that the rotational transformation matrix is, in fact, orthogonal. This fact has significance in the context of this work because it shows that an inverse transformation of the rotational transformation matrix, identically equal in this case to the transpose of the matrix, would return any point in space that was operated upon by the matrix to its original position. The matrix then can represent a valid geometrical transformation.

The rotational transformation matrix $[C]$ is defined in (3-10) as

$$[C] = \begin{pmatrix} \cos\theta\cos\phi & -\cos\theta\sin\phi & \sin\theta \\ \cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi & \cos\psi\cos\phi - \sin\theta\sin\phi\sin\psi & -\sin\psi\cos\theta \\ \sin\psi\sin\phi - \sin\theta\cos\phi\cos\psi & \sin\psi\cos\phi + \sin\theta\sin\phi\cos\psi & \cos\theta\cos\psi \end{pmatrix} \tag{A-2}$$

$[C]^T$ is then the transpose of $[C]$, defined as

$$[C]^T = \begin{pmatrix} \cos\theta\cos\phi & \cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi & \sin\psi\sin\phi - \sin\theta\cos\phi\cos\psi \\ -\cos\theta\sin\phi & \cos\psi\cos\phi - \sin\theta\sin\phi\sin\psi & \sin\psi\cos\phi + \sin\theta\sin\phi\cos\psi \\ \sin\theta & -\sin\psi\cos\theta & \cos\theta\cos\psi \end{pmatrix} \tag{A-3}$$

Let the following assignments be made:

$$\begin{array}{lll} a = \cos\psi & c = \cos\theta & e = \cos\phi \\ b = \sin\psi & d = \sin\theta & f = \sin\phi \end{array} \tag{A-4}$$

Using these assignments, the two matrices $[C]$ and $[C]^T$ become

$$[C] = \begin{pmatrix} ce & -cf & d \\ af + dbe & ae - dfb & -bc \\ bf - dea & be + dfa & ca \end{pmatrix} \tag{A-5}$$

$$[C]^T = \begin{pmatrix} ce & af + dbe & bf - dea \\ -cf & ae - dfb & be + dfa \\ d & -bc & ca \end{pmatrix} \tag{A-6}$$

Now let $[A] = [C][C]^T$ and $[B] = [C]^T[C]$

Proceeding with the matrix multiplications,

$$\begin{aligned} A_{11} &= (c^2e^2) + (c^2f^2) + (d^2) \\ A_{12} &= (acef + bcde^2) + (-acef + bcdf^2) + (-bcd) \\ A_{13} &= (bcef - acde^2) + (-bcef - acdf^2) + (acd) \\ A_{21} &= (acef + bcde^2) + (-acef + bcdf^2) + (-bcd) \\ A_{22} &= (a^2f^2 + 2abdef + b^2d^2e^2) + (a^2e^2 - 2abdef + b^2d^2f^2) + (b^2c^2) \\ A_{23} &= (abf^2 - a^2def + b^2def - abd^2e^2) + (abe^2 + a^2def - b^2def - abd^2f^2) + (-abc^2) \\ A_{31} &= (bcef - acde^2) + (-bcef - acdf^2) + (acd) \\ A_{32} &= (abf^2 + b^2def - a^2def - abd^2e^2) + (abe^2 - b^2def + a^2def - abd^2f^2) + (-abc^2) \\ A_{33} &= (b^2f^2 - 2abdef + a^2d^2e^2) + (b^2e^2 + 2abdef + a^2d^2f^2) + (a^2c^2) \end{aligned} \tag{A-7}$$

$$
\begin{aligned}
B_{11} &= (c^2e^2) + (a^2f^2 + 2abdef + b^2d^2e^2) + (b^2f^2 - 2abdef + a^2d^2e^2) \\
B_{12} &= (-c^2ef) + (a^2ef - abdf^2 + abde^2 - b^2d^2ef) + (b^2ef + abdf^2 - abde^2 - a^2d^2ef) \\
B_{13} &= (cde) + (-abcf - b^2cde) + (abcf - a^2cde) \\
B_{21} &= (-c^2ef) + (a^2ef + abde^2 - abdf^2 - d^2b^2ef) + (b^2ef - abde^2 + abdf^2 - a^2d^2ef) \\
B_{22} &= (c^2f^2) + (a^2e^2 - 2abdef + b^2d^2f^2) + (b^2e^2 + 2abdef + a^2d^2f^2) \\
B_{23} &= (-cdf) + (-abce + b^2cdf) + (abce + a^2cdf) \\
B_{31} &= (cde) + (-abcf - b^2cde) + (abcf - a^2cde) \\
B_{32} &= (-cdf) + (-abce + b^2cdf) + (abce + a^2cdf) \\
B_{33} &= (d^2) + (b^2c^2) + (a^2c^2)
\end{aligned}
\qquad \text{(A-8)}
$$

Using the trigonometric identity $\cos^2\alpha + \sin^2\alpha = 1$, then

$$
a^2 + b^2 = c^2 + d^2 = e^2 + f^2 = 1 \qquad \text{(A-9)}
$$

The equations in (A-7) and (A-8) can be simplified using these identities.

$$
\begin{aligned}
A_{11} &= c^2e^2 + c^2f^2 + d^2 = c^2(e^2 + f^2) = c^2 + d^2 = 1 \\
A_{12} &= acef + bcde^2 - acef + bcdf^2 - bcd = bcde^2 + bcdf^2 - bcd = bcd(e^2 + f^2 - 1) = 0 \\
A_{13} &= (bcef - acde^2) + (-bcef - acdf^2) + (acd) = -acde^2 - acdf^2 + acd \\
&= acd(-e^2 - f^2 + 1) = 0 \\
A_{21} &= (acef + bcde^2) + (-acef + bcdf^2) + (-bcd) = bcde^2 + bcdf^2 - bcd \\
&= bcd(e^2 + f^2 - 1) = 0 \\
A_{22} &= (a^2f^2 + 2abdef + b^2d^2e^2) + (a^2e^2 - 2abdef + b^2d^2f^2) + (b^2c^2) \\
&= a^2f^2 + b^2d^2e^2 + a^2e^2 + b^2d^2f^2 + b^2c^2 \\
&= a^2(e^2 + f^2) + b^2d^2(e^2 + f^2) + b^2c^2 = a^2 + b^2d^2 + b^2c^2 = a^2 + b^2(c^2 + d^2) = a^2 + b^2 = 1 \\
A_{23} &= (abf^2 - a^2def + b^2def - abd^2e^2) + (abe^2 + a^2def - b^2def - abd^2f^2) + (-abc^2) \\
&= ab(f^2 - d^2e^2 + e^2 - d^2f^2) - c^2 \\
&= ab[1 - d^2(e^2 + f^2) - c^2] = ab[1 - (c^2 + d^2)] = 0 \\
A_{31} &= (bcef - acde^2) + (-bcef - acdf^2) + (acd) = -acde^2 - acdf^2 + acd \\
&= acd[1 - (e^2 + f^2)] = 0 \\
A_{32} &= (abf^2 + b^2def - a^2def - abd^2e^2) + (abe^2 - b^2def + a^2def - abd^2f^2) + (-abc^2) \\
&= abf^2 - abd^2e^2 + abe^2 - abd^2f^2 - abc^2 \\
&= ab(f^2 - d^2e^2 + e^2 - d^2f^2 - c^2) = ab[1 - d^2(e^2 + f^2) - c^2] = ab[1 - (d^2 + c^2)] = 0 \\
A_{33} &= (b^2f^2 - 2abdef + a^2d^2e^2) + (b^2e^2 + 2abdef + a^2d^2f^2) + (a^2c^2) \\
&= b^2f^2 + a^2d^2e^2 + b^2e^2 + a^2d^2f^2 + a^2c^2 = b^2(e^2 + f^2) + a^2d^2(e^2 + f^2) + a^2c^2 \\
&= b^2 + a^2(c^2 + d^2) = a^2 + b^2 = 1
\end{aligned}
\qquad \text{(A-10)}
$$

$$
\begin{aligned}
B_{11} &= (c^2e^2) + (a^2f^2 + 2abdef + b^2d^2e^2) + (b^2f^2 - 2abdef + a^2d^2e^2) \\
&= c^2e^2 + a^2f^2 + b^2d^2e^2 + b^2f^2 + a^2d^2e^2 \\
&= c^2e^2 + f^2(a^2 + b^2) + d^2e^2(a^2 + b^2) = c^2e^2 + f^2 + d^2e^2 = e^2(c^2 + d^2) + f^2 = e^2 + f^2 = 1 \\
B_{12} &= (-c^2ef) + (a^2ef - abdf^2 + abde^2 - b^2d^2ef) + (b^2ef + abdf^2 - abde^2 - a^2d^2ef) \\
&= -c^2ef + a^2ef - b^2d^2ef + b^2ef - a^2d^2ef = -c^2ef + ef(a^2 + b^2) - d^2ef(a^2 + b^2) \\
&= ef[1 - (c^2 + d^2)] = 0 \\
B_{13} &= (cde) + (-abcf - b^2cde) + (abcf - a^2cde) = cde - b^2cde - a^2cde \\
&= cde[1 - (a^2 + b^2)] = 0 \\
B_{21} &= (-c^2ef) + (a^2ef + abde^2 - abdf^2 - d^2b^2ef) + (b^2ef + abde^2 + abdf^2 - a^2d^2ef) \\
&= -c^2ef + a^2ef - d^2b^2ef + b^2ef - a^2d^2ef = ef(a^2 + b^2) - d^2ef(a^2 + b^2) - c^2ef \\
&= ef[1 - (c^2 + d^2)] = 0 \\
B_{22} &= (c^2f^2) + (a^2e^2 - 2abdef + b^2d^2f^2) + (b^2e^2 + 2abdef + a^2d^2f^2) \\
&= c^2f^2 + a^2e^2 + b^2d^2f^2 + b^2e^2 + a^2d^2f^2 = c^2f^2 + e^2(a^2 + b^2) + d^2f^2(a^2 + b^2) \\
&= e^2 + f^2(c^2 + d^2) = e^2 + f^2 = 1 \\
B_{23} &= (-cdf) + (-abce + b^2cdf) + (abce + a^2cdf) = -cdf + b^2cdf + a^2cdf \\
&= -cdf + b^2cdf + a^2cdf = cdf(a^2 + b^2) - cdf = 0 \\
B_{31} &= (cde) + (-abcf - b^2cde) + (abcf - a^2cde) = cde - b^2cde - a^2cde \\
&= cde - b^2cde - a^2cde = cde[1 - (a^2 + b^2)] = 0 \\
B_{32} &= (-cdf) + (-abce + b^2cdf) + (abce + a^2cdf) = -cdf + b^2cdf + a^2cdf \\
&= cdf[(a^2 + b^2) - 1] = 0 \\
B_{33} &= (d^2) + (b^2c^2) + (a^2c^2) = d^2 + b^2c^2 + a^2c^2 = d^2 + c^2(a^2 + b^2) = c^2 + b^2 = 1
\end{aligned}
\tag{A-11}
$$

From (A-10) and (A-11),

$$
\begin{aligned}
&A_{11} = B_{11} = 1 \quad A_{12} = B_{11} = 0 \quad A_{13} = B_{11} = 0 \\
&A_{21} = B_{21} = 0 \quad A_{22} = B_{22} = 1 \quad A_{23} = B_{23} = 0 \\
&A_{31} = B_{31} = 0 \quad A_{32} = B_{32} = 0 \quad A_{33} = B_{33} = 1
\end{aligned}
\tag{A-12}
$$

It is clear then that

$$
[A] = [B] = I_3 \quad \Leftrightarrow \quad [C][C]^T = [C]^T[C] = I_3 \tag{A-13}
$$

The matrix $[C]$ is therefore **orthogonal.**

QED

An alternate proof of the orthogonality of the rotational transformation matrix, i.e. that equation (A-1) is satisfied can be based upon the fact that

$$[C] = [E][F][G] \tag{A-14}$$

where $[E]$, $[F]$ and $[G]$ represent the three matrices on the right hand side of (3-10). It is easy to show that these matrices are orthogonal.

$$[C] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A-15}$$

$$[E][E]^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} = I_3 \tag{A-16}$$

$$[F][F]^T = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} = I_3 \tag{A-17}$$

$$[G][G]^T = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} = I_3 \tag{A-18}$$

Then using matrix properties from linear algebra:

$$([E][F][G])^T([E][F][G]) = [G]^T[F]^T[E]^T[E][F][G] \tag{A-19}$$

$$[G]^T[F]^T[E]^T[E][F][G] = [G]^T[F]^T([E]^T[E])[F][G] = [G]^T[F]^T[F][G] \tag{A-20}$$

$$[G]^T[F]^T[F][G] = [G]^T([F]^T[F])[G] = [G]^T[G] = I_3 \tag{A-21}$$

The commuted expression is shown similarly.

QED

Although this proof is straightforward, and vastly simpler than the proof outlined in (A-2) through (A-13), the former proof is useful because it provides an independent verification of the symbolic correctness of (A-2).

Appendix B

DERIVATION OF JACOBIAN FOR SQUARED DISTANCE FUNCTION

The Jacobian of the squared distance function which is used as the objective function in the unconstrained localization problem of Chapter 3 is estimated in the NAG routine E04JAF by difference quotients. Because it is expected that another minimization routine could make productive use of a symbolic Jacobian, the Jacobian for the squared distance function is presented here. Such a routine could be the routine E04UCF used in Chapter 4, with application to the unconstrained localization problem for enhanced precision and efficiency. The derivation is analogous to that of the oriented distance function which was used as the constraint function in the constrained localization problem of Chapter 4.

Recalling the notation of Chapters 3 and 4, assume again a parametric design surface, $\mathbf{P}(u,v)$, a set of m measured points $\mathbf{R}_i$, and the set of nearest points or projections, $\mathbf{Q}_i$. The minimum distance from a measured point $\mathbf{R}_i$ to the design surface $\mathbf{P}(u,v)$ is defined as

$$d(\mathbf{R}_i, \mathbf{Q}_i) = |\, \mathbf{R}_i - \mathbf{Q}_i \,| = \min_{u,v} d\,[\mathbf{R}_i, \mathbf{P}(u,v)] \qquad \text{(B-1)}$$

A new set of transformed points $\mathbf{r}_i$, can be obtained from the original set of measured points $\mathbf{R}_i$, by the following operation:

$$\mathbf{r}_i = [C]\mathbf{R}_i + \mathbf{t} \qquad \text{(B-2)}$$

If $\mathbf{r}_i$ is a transformed point given by equation (B-2), and $\mathbf{q}_i$ is the projection of the transformed point onto the design surface, then the **transformed minimum squared distance** from $\mathbf{r}_i$ to $\mathbf{q}_i$, $[d(\mathbf{r}_i, \mathbf{q}_i)]^2$, may be defined as

$$[d(\mathbf{r}_i, \mathbf{q}_i)]^2 = (\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i) \tag{B-3}$$

The Jacobian for the squared distance function is the set of first partial derivatives of the squared distance function in the six independent variables of the problem. Since the transformed minimum squared distance function is the objective function for the localization problem during any particular iteration step, the determination of the Jacobian for this function consists of calculating a set of m first partial derivatives in the six variables φ, θ, ψ, t_x, t_y, and t_z. The Jacobian $\boldsymbol{J}_i$, $1 \le i \le m$, for this problem may thus be defined as

$$\boldsymbol{J}_i^T = \begin{pmatrix} \dfrac{\partial[d(\mathbf{r}_i, \mathbf{q}_i)]^2}{\partial\phi} \\ \dfrac{\partial[d(\mathbf{r}_i, \mathbf{q}_i)]^2}{\partial\theta} \\ \dfrac{\partial[d(\mathbf{r}_i, \mathbf{q}_i)]^2}{\partial\psi} \\ \dfrac{\partial[d(\mathbf{r}_i, \mathbf{q}_i)]^2}{\partial t_x} \\ \dfrac{\partial[d(\mathbf{r}_i, \mathbf{q}_i)]^2}{\partial t_y} \\ \dfrac{\partial[d(\mathbf{r}_i, \mathbf{q}_i)]^2}{\partial t_z} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial(\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial\phi} \\ \dfrac{\partial(\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial\theta} \\ \dfrac{\partial(\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial\psi} \\ \dfrac{\partial(\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial t_x} \\ \dfrac{\partial(\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial t_y} \\ \dfrac{\partial(\mathbf{r}_i - \mathbf{q}_i) \cdot (\mathbf{r}_i - \mathbf{q}_i)}{\partial t_z} \end{pmatrix} \tag{B-4}$$

which when the partial derivatives are expanded is equivalent to

$$
J_i^T = \begin{pmatrix}
2\left(\frac{\partial[C]}{\partial\phi}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial\phi} - \frac{\partial \mathbf{q}_i}{\partial\phi}\right)\cdot(\mathbf{r}_i - \mathbf{q}_i) \\
2\left(\frac{\partial[C]}{\partial\theta}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial\theta} - \frac{\partial \mathbf{q}_i}{\partial\theta}\right)\cdot(\mathbf{r}_i - \mathbf{q}_i) \\
2\left(\frac{\partial[C]}{\partial\psi}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial\psi} - \frac{\partial \mathbf{q}_i}{\partial\psi}\right)\cdot(\mathbf{r}_i - \mathbf{q}_i) \\
2\left(\frac{\partial[C]}{\partial t_x}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial t_x} - \frac{\partial \mathbf{q}_i}{\partial t_x}\right)\cdot(\mathbf{r}_i - \mathbf{q}_i) \\
2\left(\frac{\partial[C]}{\partial t_y}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial t_y} - \frac{\partial \mathbf{q}_i}{\partial t_y}\right)\cdot(\mathbf{r}_i - \mathbf{q}_i) \\
2\left(\frac{\partial[C]}{\partial t_z}\mathbf{R}_i + \frac{\partial \mathbf{t}}{\partial t_z} - \frac{\partial \mathbf{q}_i}{\partial t_z}\right)\cdot(\mathbf{r}_i - \mathbf{q}_i)
\end{pmatrix} \tag{B-5}
$$

It is clear that

$$
\frac{\partial \mathbf{t}}{\partial\phi} = \frac{\partial \mathbf{t}}{\partial\theta} = \frac{\partial \mathbf{t}}{\partial\psi} = 0 \quad \text{and} \quad \frac{\partial[C]}{\partial t_x} = \frac{\partial[C]}{\partial t_y} = \frac{\partial[C]}{\partial t_z} = [0] \tag{B-6}
$$

As developed in Chapter 4, by assumption, $\mathbf{q}_i$ is an orthogonal projection of $\mathbf{r}_i$ on $\mathbf{P}(u, v)$ and $\mathbf{n}_i$ is the unit normal vector to the surface $\mathbf{P}(u, v)$ at the projection $\mathbf{q}_i$. Therefore the vector $(\mathbf{r}_i - \mathbf{q}_i)$ is collinear with the unit normal vector $\mathbf{n}_i$ and the following relationships exist:

$$
\frac{\partial \mathbf{q}_i}{\partial\phi}\cdot(\mathbf{r}_i - \mathbf{q}_i) = \frac{\partial \mathbf{q}_i}{\partial\theta}\cdot(\mathbf{r}_i - \mathbf{q}_i) = \frac{\partial \mathbf{q}_i}{\partial\psi}\cdot(\mathbf{r}_i - \mathbf{q}_i) = 0 \tag{B-7}
$$

$$
\frac{\partial \mathbf{q}_i}{\partial t_x}\cdot(\mathbf{r}_i - \mathbf{q}_i) = \frac{\partial \mathbf{q}_i}{\partial t_y}\cdot(\mathbf{r}_i - \mathbf{q}_i) = \frac{\partial \mathbf{q}_i}{\partial t_z}\cdot(\mathbf{r}_i - \mathbf{q}_i) = 0 \tag{B-8}
$$

By using the expressions of (B-6) through (B-8) the Jacobian $\boldsymbol{J}_i$ of (B-5) may be simplified to

$$J_i^T = \begin{pmatrix} 2\dfrac{\partial[C]}{\partial\phi}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial[C]}{\partial\theta}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial[C]}{\partial\psi}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial \mathbf{t}}{\partial t_x} \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial \mathbf{t}}{\partial t_y} \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial \mathbf{t}}{\partial t_z} \cdot (\mathbf{r}_i - \mathbf{q}_i) \end{pmatrix} \tag{B-9}$$

and this may be finally simplified to

$$J_i^T = \begin{pmatrix} 2\dfrac{\partial[C]}{\partial\phi}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial[C]}{\partial\theta}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial[C]}{\partial\psi}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\mathbf{e}_x^i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\mathbf{e}_y^i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\mathbf{e}_z^i \cdot (\mathbf{r}_i - \mathbf{q}_i) \end{pmatrix} = \begin{pmatrix} 2\dfrac{\partial[C]}{\partial\phi}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial[C]}{\partial\theta}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2\dfrac{\partial[C]}{\partial\psi}\mathbf{R}_i \cdot (\mathbf{r}_i - \mathbf{q}_i) \\ 2(r_x^i - q_x^i) \\ 2(r_y^i - q_y^i) \\ 2(r_z^i - q_z^i) \end{pmatrix} \tag{B-10}$$

In this expression $\mathbf{e}_x^i$, $\mathbf{e}_y^i$ and $\mathbf{e}_z^i$ are unit vectors along the axes of the coordinate system which are parallel to the components of the translation vector t. The three differences r_x^i-q_x^i, r_y^i-q_y^i and r_z^i-q_z^i are the scalar components of the difference vector $\mathbf{r}_i$-$\mathbf{q}_i$. The rotational elements of the Jacobian require calculating the first partial derivatives of the rotational transformation matrix $[C]$ in each of the variables φ, θ and ψ. Those partial derivatives are given by the following three matrix equations.

$$\frac{\partial[C]}{\partial\phi} = \begin{pmatrix} -\cos\theta\sin\phi & -\cos\theta\cos\phi & 0 \\ \cos\psi\cos\phi - \sin\theta\sin\psi\sin\phi & -\cos\psi\sin\phi - \sin\theta\cos\phi\sin\psi & 0 \\ \sin\psi\cos\phi + \sin\theta\sin\phi\cos\psi & -\sin\psi\sin\phi + \sin\theta\cos\phi\cos\psi & 0 \end{pmatrix} \tag{B-11}$$

$$\frac{\partial[C]}{\partial\theta} = \begin{pmatrix} -\sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \\ \cos\theta\sin\psi\cos\phi & -\cos\theta\sin\phi\sin\psi & \sin\psi\sin\theta \\ -\cos\theta\cos\phi\cos\psi & \cos\theta\sin\phi\cos\psi & -\sin\theta\cos\psi \end{pmatrix} \tag{B-12}$$

$$\frac{\partial[C]}{\partial\psi} = \begin{pmatrix} 0 & 0 & 0 \\ -\sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi & -\sin\psi\cos\phi - \sin\theta\sin\phi\cos\psi & -\cos\psi\cos\theta \\ \cos\psi\sin\phi + \sin\theta\cos\phi\sin\psi & \cos\psi\cos\phi - \sin\theta\sin\phi\sin\psi & -\cos\theta\sin\psi \end{pmatrix} \tag{B-13}$$

The rotational elements of the Jacobian are then determined by multiplying each of the three matrices of (B-10), (B-11) and (B-12) by $\mathbf{R}_i$ and substituting the results into (B-10).

The Jacobian of (B-7) was not used in this investigation because the routine that was implemented for the unconstrained localization problem could not use gradients supplied by the user. It is expected that the use of this Jacobian would greatly improve the efficiency and accuracy of the optimization process, and it is clear from this development that the Hessian and other higher order partial derivatives could be readily derived.

Appendix C

DETERMINATION OF PARAMETERS IN DESIGN SURFACE FOR HIGH ACCURACY MINIMUM DISTANCE CALCULATION

The very accurate determination of minimum distance from a point to a parametric surface that is used in the oriented distance function of Chapter 4 uses the u and v parameters in the design surface of the orthogonal projection of the given point onto the design surface. Therefore the problem is to find as accurately as possible the pair of u, v parameters which correspond to the projection of a given measured point onto the design surface.

In Chapter 3 a modified Newton algorithm (NAG routine E04KCF) was used to determine the minimum distance from a measured point to the design surface for use in the squared distance function. This method provides a fast and reasonably accurate determination of the minimum distance by finding a minimum of the squared distance of the measured point from an arbitrary point P(u,v) with parameters u,v in the vicinity of a starting point approximation (u_0, v_0). Given the inherent inaccuracies of any minimization routine in floating point arithmetic, the solution will be only approximate.[23]

An improvement in the accuracy of the calculation of minimum distance from a measured point to the design surface can be achieved by using the orthogonality of the projection to determine the values of the parameters u and v in the design surface. If the values of u and v from the minimization routine are used as a starting point, then the

[23] Using the NAG routine E04KCF in 16 digit floating point arithmetic, only 7 digits of precision in the calculation of the paramters u and v for the minimum distance calculation can typically be obtained.

orthogonality condition, which by an assumption in section 3.2 must exist at the projection of the measured point onto the design surface, provides a simple method for obtaining a very accurate value for the minimum distance. The development of the method is similar to that presented in Appendix C of [Kriezis 90].

Using the notation of Chapters 3 and 4, consider again the minimum distance $d(\mathbf{R}_i, \mathbf{Q}_i)$, from a measured point $\mathbf{R}_i$, to the design surface $\mathbf{P}(u,v)$. Since $\mathbf{Q}_i$ is defined as the projection of $\mathbf{R}_i$ onto the design surface $\mathbf{P}(u,v)$, orthogonality necessarily requires that

$$(\mathbf{R}_i - \mathbf{Q}_i) \cdot \partial_u \mathbf{P}(u,v) = 0 \tag{C-1}$$

and

$$(\mathbf{R}_i - \mathbf{Q}_i) \cdot \partial_v \mathbf{P}(u,v) = 0 \tag{C-2}$$

where

$$\mathbf{Q}_i = \mathbf{P}(u,v) \tag{C-3}$$

Given the necessary and sufficient conditions for orthogonality of (C-1) and (C-2), the determination of the u and v parameters in the design surface $\mathbf{P}(u,v)$ which correspond to the projection point $\mathbf{Q}_i$ can be accomplished by finding the zeros in u and v which satisfy the conditions of (B-1) and (B-2). So consider two functions $\boldsymbol{F}_1(u,v)$ and $\boldsymbol{F}_2(u,v)$, defined as

$$F_1(u,v) = (\mathbf{R}_i - \mathbf{Q}_i) \cdot \partial_u \mathbf{P}(u,v) = (\mathbf{R}_i - \mathbf{P}(u,v)) \cdot \partial_u \mathbf{P}(u,v) \tag{C-4}$$

and

$$F_1(u,v) = (\mathbf{R}_i - \mathbf{Q}_i) \cdot \partial_v \mathbf{P}(u,v) = (\mathbf{R}_i - \mathbf{P}(u,v)) \cdot \partial_v \mathbf{P}(u,v) \tag{C-5}$$

The zeros in u and v for functions $F_1(u,v)$ and $F_2(u,v)$ are found using the NAG routine C05PBF which utilizes another modified Newton method to find the zeros of multivariable functions with the gradients in each variable supplied by the user.

The gradients for $F_1(u,v)$ and $F_2(u,v)$ can be expressed as

$$\partial_u F_1 = -\partial_u \mathrm{P}(u,v) \cdot \partial_u \mathrm{P}(u,v) + (\mathrm{R}_i - \mathrm{P}(u,v)) \cdot \partial_{uu} \mathrm{P}(u,v) \tag{C-6}$$

$$\partial_u F_2 = -\partial_u \mathrm{P}(u,v) \cdot \partial_v \mathrm{P}(u,v) + (\mathrm{R}_i - \mathrm{P}(u,v)) \cdot \partial_{uv} \mathrm{P}(u,v) \tag{C-7}$$

$$\partial_v F_1 = -\partial_v \mathrm{P}(u,v) \cdot \partial_u \mathrm{P}(u,v) + (\mathrm{R}_i - \mathrm{P}(u,v)) \cdot \partial_{vu} \mathrm{P}(u,v) \tag{C-8}$$

$$\partial_u F_2 = -\partial_v \mathrm{P}(u,v) \cdot \partial_v \mathrm{P}(u,v) + (\mathrm{R}_i - \mathrm{P}(u,v)) \cdot \partial_{vv} \mathrm{P}(u,v) \tag{C-9}$$

There is obviously a time penalty associated with the use of this method rather than the simple minimization routine. The improvement in accuracy may justify the use of this method when accuracy is more important than computational speed. In particular, this method was employed in the oriented distance function computation used for the constrained localization problem of Chapter 4.

References

[Alt 88] Alt, H., Mehlhorn, K., Wagener, H., Welzl, E.
Congruence, Similarity, and Symmetries of Geometric Objects.
Discrete Computational Geometry 3:237-256, 1988.

[Bardis 91] Bardis, L., Jinkerson, R. A., Patrikalakis, N. M.
Localization for Automated Inspection of Curved Surfaces.
Transactions of the First International Offshore and Polar Engineering Conference, Edinburgh, Scotland, UK. August 1991. Golden, CO: ISOPE.
(To appear).

[Bourdet 88] Bourdet, P., Clement, A.
A Study of Optimal-Criteria Identification Based on the Small-Displacement Screw Model.
Annals of the CIRP 37(1):503-506, January, 1988.

[Gill 74] Gill, P. E., Murray, W.
Newton-Type Methods for Unconstrained and Linearly Constrained Optimization.
Mathematical Programming 7:311-350, 1974.

[Gill 76] Gill, P. E., Murray, W.
Minimization Subject to Bounds on the Variables.
National Physical Laboratory Report NAC 72, 1976.

[Gill 86] Gill, P. E., Hammarling, S. J., Saunders, M. A., Wright, M. H.
User's Guide for LSSOL (Version 1.0)
Department of Operations Research, Stanford University
Technical Report SOL 86-6R, 1986.

[Gordon 74] Gordon W. J., Riesenfeld, R. F.
B-Spline Curves and Surfaces.
Computer Aided Geometric Design, pages 95-126, Edited by Barnhill, R. E., and Riesenfeld, R. F., Academic Press, Inc., 1974.

[Gunnarsson 87a] Gunnarsson, K. T.
Optimal Part Localization by Data Base Matching with Sparse and Dense Data.
PhD thesis, Carnegie-Mellon University, 1987.

[Gunnarsson 87b] Gunnarsson, K. T., Prinz, F. B.
CAD Model-Based Localization of Parts in Manufacturing.
Computer, Journal of the Computer Society of the IEEE :66-74, August, 1987.

[Holter 90] Holter, N.
Private communication, October 15, 1990.

[Hottel 91] Hottel, G. R., Tuohy, S. T., Alourdas, P. G., Patrikalakis, N.M.
Praxiteles: A Geometric Modeling and Interrogation System.
MIT Ocean Engineering Design Laboratory Memorandum 91-6, Cambridge, MA, March 14, 1991.

[Imai 88] Imai, K., Sumino, S., Imai, H.
Minimax Geometric Fitting of Two Corresponding Sets of Points.
Proceedings of the 5th Annual Symposium on Computational Geometry.
Saarbruecken, Germany, June 5-7, 1989.

[Jinkerson 91] Jinkerson, R. A., Abrams, S. L., Wolter, F. E., Patrikalakis, N. M.
Unconstrained Localization Program User's Manual, Version 1
MIT Ocean Engineering Design Laboratory Memorandum 91-8a, Cambridge, MA, April 26, 1991

[Koehler 91] Koehler, M., York, T.
Private communication, January 1991.

[Kriezis 90] Kriezis, G. A.
Algorithms for Rational Spline Surface Intersections.
PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, March, 1990.

[Kriezis 91] Kriezis, G. A., Patrikalakis, N. M., Wolter, F. E.
Topological and Differential Equation Methods for Surface Intersections
Computer-Aided Design, 1991.
(To appear).

[NAG 89] *NAG Fortran Library Manual*
Mark 13 edition, Numerical Algorithms Group, Oxford, England, 1989.

[Patrikalakis 90] Patrikalakis, N. M., Bardis, L.
Localization of Rational B-Spline Surfaces.
Engineering with Computers, 1991.
(To appear).

[Pegna 90] Pegna, J., Wolter, F. E.
Designing and Mapping Trimming Curves on Surfaces Using Orthogonal Projection.
In *Advances in Design Automation 1990, Volume One, Computer Aided and Computational Design,*
ASME DE-Vol. 23-1:235-245, 1990.

[Thorne 85] Thorne. H. F., Prinz, F. B., Kirchner, H. O. K.
Robotic Inspection by Database Matching.
Technical Report CMU-RI-TR-85-4, The Robotics Institute, Carnegie Mellon University, March, 1985.

[Tuohy 91] Tuohy, S. T., Patrikalakis, N. M.
Geometric Representation of Marine Propulsors.
MIT Ocean Engineering Design Laboratory Memorandum 91-7, Cambridge, MA, April 9, 1991.

[Wolter 85] Wolter, F. E.
Cut Loci in Bordered and Unbordered Riemannian Manifolds.
PhD thesis, Technical University of Berlin, Department of Mathematics, December 1985.

Thesis
J5414
c.1

Jinkerson
Constrained and unconstrained localization for automated inspection of marine propellers.